

明 細 書

演算処理装置、演算処理装置設計方法および論理回路設計方法

5 技術分野

この発明は、演算処理技術に関し、更に詳しくは演算 L S I 等の高速化や低消費電力化などを実現する符号化に基づいた演算処理方式とその符号化の設計方法および演算処理装置に関する。

10 背景技術

演算 L S I は画像圧縮伸張、コンピュータグラフィック、画像認識、音声認識など、近年非常に多くの分野に使われている。例えば画像圧縮伸張の M P E G 2、M P E G 4 などでは、圧縮時も伸張時も多量の演算が必要であり、L S I に占める演算部の割合は極めて大きい。

15 画像処理に限らずコンピュータなどの数値演算においても要求される演算規模は増大の一途にあり、高速化に対するニーズは限りないものがある。また演算器内部では一般にトランジスタのスイッチングのオン・オフが頻繁に行われる為、消費電力の問題が無視できない重要な課題となる。

つまり演算に対して、(1) さらに高速化を図りたい、(2) 消費電力を削減したいという、2つの強いニーズが存在する。

このようなニーズに対して、微細加工技術やデバイス特性の改善などのアプローチは着実に進められている。しかし、このようなデバイスによるアプローチではなしに演算器そのものを改善するアプローチが考えられる。

25 演算器そのものに対するアプローチの中で、純粋に論理的なアプローチで高速化などの高性能化を図る方式として、本出願の発明者達は、“高速演算方式”なる提案を行っている(国際公開 W O 9 1 / 1 0 1 8 6 すなわち特開平 3 - 2 0 1 1 1 4 号公報。以下、関連特許文献という)。

この提案の概要を、関連特許文献の表記に従い以下に示す。

P 個の二項演算 $A_p : \Omega \times \Omega \rightarrow \Omega$ ($p=1,2,\dots,P$) と Q 個の単項演算 $B_q : \Omega \rightarrow \Omega$ ($q=1,2,\dots,Q$) が定義され、これらの演算に関して閉じた有限集合 Ω を考える。 Ω , A_p , B_q を合わせ旧演算系 (Ω, A_p, B_q) と呼ぶ。旧演算系の演算 A_p , B_q を実行する代わりに、集合 Ω 上のデータを一度符号器 $\Phi : \Omega \rightarrow \Omega'$ により

5 $|\Omega'| \geq |\Omega|$ ($|\Omega'|, |\Omega|$ はそれぞれ Ω', Ω の要素数) なる集合 Ω' 上のデータに変換し、演算 A_p , B_q に対応した Ω' 上の演算 $A'_p : \Omega' \times \Omega' \rightarrow \Omega'$ と $B'_q : \Omega' \rightarrow \Omega'$ を実行し、 Ω' 上のデータとして得られた演算結果を復号器 $\Psi : \Omega' \rightarrow \Omega$ で変換することにより、本来計算しようとした Ω 上の演算結果を得る。この演算方式で、

$\Omega', A'_p, B'_q, \Phi, \Psi$ を合わせ新演算系 $(\Omega', A'_p, B'_q, \Phi, \Psi)$ と呼び、新演算系と旧演算系との間に以下の①②③④の 4 式を課すことで、符号化条件 (旧演算系で計算する結果と新演算系で計算する結果が一致することを保証させる条件) を満足させる。

$$\Phi(X) \in [X] \subset \Omega' \quad (\text{for } \forall X \in \Omega) \quad ①$$

$$\Psi([X]) = X \in \Omega \quad (\text{for } \forall X \in \Omega) \quad ②$$

$$Z = A_p(X, Y) \Leftrightarrow [Z] \supset A'_p([X], [Y]) \quad (\text{for } \forall X, Y, Z \in \Omega, \text{ 全ての } p) \quad ③$$

$$15 \quad Y = B_q(X) \Leftrightarrow [Y] \supset B'_q([X]) \quad (\text{for } \forall X, Y \in \Omega, \text{ 全ての } q) \quad ④$$

ここに $[X]$ は Ω 上の任意の要素 X に対応した Ω' 上の部分集合で、集合族

$\{[X]\}_{X \in \Omega}$ を符号と呼び、 $C = \bigcup_{X \in \Omega} [X]$ は符号の全体集合、 $N_C = \Omega' - C$ は非符号の集合である。

また①②より

$$20 \quad [X] \neq \phi \quad (\phi \text{ は空集合}) \quad ⑤$$

$$X \neq Y \Leftrightarrow [X] \cap [Y] = \phi \quad ⑥$$

が成り立つものである。

以上のような①乃至④の符号化条件を満足し、新演算系の演算器が極力簡単になるように、符号器、復号器、新演算器をコンピュータを利用して決定し、その
25 ことで高速化などを達成するというのが前記の関連特許文献における提案であった。

この先の提案は、加減乗除の四則演算に限らず、一般的な演算や、演算とみなせる写像一般に対して考案されたものであるが、以下のような大きく6つの問題があった。

- i) 旧演算系、新演算系のデータの集合が一般的な有限集合 Ω 、 Ω' となっている為、演算器、符号器、復号器を表す写像も一般的な写像で表され、論理式との対応が不明解である。
 - ii) 旧演算系が Ω に関して閉じた演算系だけを対象としている為、一般的な演算が対象外である。
 - iii) 符号化の関係式が上記一般的写像と集合により示されている為、関係を満足することの論理的な意味が不明解であり、その為に関係式を満足する実際の論理式を導くことが困難な場合がある。
 - iv) 回路の簡単化の要請を定式化できていない。
 - v) 符号化の具体例が非冗長符号の場合しか示されていない。
 - vi) 符号化演算方式としてのシステム構成の具体化が不十分である。
- 9 1 / 1 0 1 8 6 (特開平 3 - 2 0 1 1 1 4 号公報)

発明の開示

- この発明は、以上の事情を考慮してなされたものであり、以上の関連特許文献における問題点を解決する為に、与えられた演算を必要があれば拡張することにより、旧演算系の全ての演算と新演算系の全ての演算および符号器、復号器を全て多入力多出力の r 値論理表現で記述することで、 r 値論理による具体的な論理設計を可能とし、また2値論理表現における新たな表現形式を用いて、論理式による方程式の形で定式化することで、これらの符号化の条件などを満足させた新演算系の効率的な論理設計を可能とし、回路規模や演算遅延時間を大幅に削減した符号化演算方式に基づく回路設計を実現し、LSIに実装が可能な符号化演算装置を提供することを目的としている。

この発明によれば、上述の目的を達成するために、特許請求の範囲に記載のとおり構成を採用している。

以下本発明の各側面を順に説明する。

(1) 発明の第1の側面

関連特許文献の問題 i) ii) を解決するために、この発明の第1の側面によれば、与えられた演算系(元演算系と呼ぶことにする。)の演算器を必要があれば拡張して r 値論理の写像として扱い、演算的に閉じた旧演算系の演算器を構成し、
 5 且つ新演算系の符号器、復号器および演算器を全て r 値論理の写像として扱い、
 関連特許文献に示された符号化条件式または特徴論理関数により表される方程式の形の符号化条件式または元演算系と演算器と新演算系の演算器の入出力のトポロジーが同じになることを符号化の条件として満足するようにして、新演算系の r 値論理による論理設計を可能にする。

10 (2) 発明の第2の側面

発明の第2の側面によれば、集合の特徴関数を2値の論理関数として扱い、特徴関数間の関係や写像の合成などを見通し良く且つ効率的に扱うための、新たに考案した母関数という写像の表現法を導入し、また各種の関係式を導き、論理回路設計を行う。

15 (3) 発明の第3の側面

発明の第3の側面によれば、論理関数の簡単さを変数依存度に対応した方程式の形で扱うことで簡素の構成の論理関数を設計する。

発明の第2の側面および第3の側面を総合的に適用することもできる。

(4) 発明の第4の側面

20 関連特許文献の問題 iii) を解決するために、この発明の第4の側面によれば、旧演算系の演算器および新演算系の符号器、復号器および演算器の全てを2値論理の論理関数として扱い、これらを前記母関数により表現し、母関数で表現された符号化条件式を導き、新演算系の論理設計を見通し良く且つ効率的に行うことを可能にする。

25 (5) 発明の第5の側面

関連特許文献の問題 iv) を解決するために、この発明の第5の側面によれば、新演算系の簡単化の条件を方程式の形であたえ、前記第2の側面に従う符号化条件と合わせて論理設計を行うことで、簡単化の条件を満足する新演算系の論理設計を可能にする。

(6) 発明の第6の側面

関連特許文献の問題 iii) と iv) を同時に解決するために、この発明の第6の側面によれば、符号器を同型写像に限定し、上述の第2から第4の側面に従って論理設計を行うことで、符号化条件と簡単化条件を満足する新演算系の効率的な論理設計を可能にする。

尚、上述の発明の第2の側面の手法は、符号化演算方式に限らず論理設計や論理解析などに広く用いることができる発明である。

また関連特許文献の問題 v) を解決するための本発明の具体的な構成は、上述の各側面の符号化演算方式および設計方法に基づき論理設計された実施例により示す。

[記号と記法]

以下に本発明を詳細に説明するが、その準備として説明に用いる記号、記法に就いて定めて置く。

r 値の集合は B_r 、基数 r 、語長 n の数表現の集合は B_r^n (B_r の n 個の直積) で表し、2 値の場合は $B_2 = B = \{0,1\}$ として B を用いる。 B_r の要素はスカラーと考え、 $x, y, \dots \in B_r$ のように小文字 x, y, \dots で、 B_r^n の要素は n 成分のベクトルと考え、 $X, Y, \dots \in B_r^n$ のように大文字で扱う。(例えば X は $X = (x_1, x_2, \dots, x_n)$)

n 変数の論理関数はスカラー関数と考え $f(X) = f(x_1, x_2, \dots, x_n)$ の f のように小文字を用い、またこれを $f: B_r^n \rightarrow B_r$ なる写像と考え同じ記号を用いる。

m 個の n 変数 r 値論理関数 $f_j(X)$ ($j=1, 2, \dots, m$) をまとめてベクトル関数として扱うときには $F(X) = (f_1(X), f_2(X), \dots, f_m(X))$ の F のように大文字を用い、またこれを $F: B_r^n \rightarrow B_r^m$ なる写像と考え同じ記号を用いる。

2 値の場合には変数 (または定数) の否定は \bar{x} で表し、2 つの変数 (または定数) x, y の論理積 (and) は $x \cdot y$ または xy のように表し、 x, y の論理和 (or)

は $x \cup y$ で表し、 n 個の変数 (または定数) $x_i (i=1, 2, \dots, n)$ の論理積は $\bigcap_i x_i$ で、論

理和は $\bigcup_i x_i$ で表し、 i の和や積の範囲指定がない場合には全体で積、和をとるものとする。

同様に例えば $\bigcap_X f(X)$ は $X \in B^n$ ならば B^n 全体についての積を意味する。和も同様である。

2 変数（または定数） x, y のブール環の和（e x o r）は $x + y$ と表し、和の範囲は前記と同様に扱う。

- 5 x^a を $a=1$ の時 $x^a = x$ 、 $a=0$ の時 $x^a = \bar{x}$ なるものと定義すると、 $x^a = xa \cup \bar{x}\bar{a}$ と表せる。これを用い $X, A \in B^n$ に対し $X^A = \bigcap_{i=1}^n x_i^{a_i} = \bigcap_{i=1}^n (x_i \cdot a_i \cup \bar{x}_i \cdot \bar{a}_i)$

と定義する。この時 n 変数 x_1, x_2, \dots, x_n からなる最小項は X^A ($X, A \in B^n$) なる形で表せる。例えば $n=3$ の時には、 $A=(0,1,0)$ の時には X^A は $\bar{x}_1 \cdot x_2 \cdot \bar{x}_3$ を表す。また $A=(0,0,0), (0,0,1), \dots, (1,1,1)$ に数字の $A=0, 1, \dots, 7$ を対応させ、最小項を

- 10 X^0, X^1, \dots, X^7 のように略記することもある。

また X^A を用いると、ブール関数 $f(X)$ の積和標準形は $f(X) = \bigcup_A f(A) \cdot X^A$ と表せる。

また 2 値論理関数 $f(X)$ の変数 x_i によるブール微分は $f(X)$ の x_i に 0, 1 を代入した式のブール環の和、つまり $\frac{df(X)}{dx_i} = f(x_0, x_1, \dots, 0, \dots, x_n) + f(x_0, x_1, \dots, 1, \dots, x_n)$

- 15 と表される。 $\frac{df(X)}{dx_i} = 0$ の時、関数 $f(X)$ は変数 x_i に依存していない。

また一般に次が成り立つ。 $a_i = 0 \Leftrightarrow \bigcup_i a_i = 0$ 、 $f(X) = 0 \Leftrightarrow \bigcup_X f(X) = 0$

尚、集合に対する和集合と共通部分についても混乱は無いと思われるので通常通り、 \cup, \cap を用いる。

- 20 [発明の構成]

以下、この発明を詳細に説明する。

[発明の第 1 の側面]

発明の第1の側面は請求項1～4の発明に対応する。

前記の関連特許文献のように閉じた演算系が定義された有限集合 Ω に限定しない。つまり演算の入力データの集合 Ω_{in} と演算の出力空間の集合 Ω_{out} が $\Omega_{in} \neq \Omega_{out}$ の場合も対象にする。

5 具体的には Q 種の単項演算 $G^q: \Omega_{G^q, in} \rightarrow \Omega_{G^q, out}$ ($q=1,2,\dots,Q$) および P 種の

2項演算 $F^p: \Omega_{F^p, in} \times \Omega_{F^p, in} \rightarrow \Omega_{F^p, out}$ ($p=1,2,\dots,P$)および S 種の T 項演算

$H^s: \Omega_{H^s, in} \times \Omega_{H^s, in} \times \dots \times \Omega_{H^s, in} \rightarrow \Omega_{H^s, out}$ ($s=1,2,\dots,S$) が1つまたは複数個 ($Q+P+S \geq 1$) 定義された演算系(元演算系)に対して符号化演算方式を考える。

それぞれの入力空間の有限集合 $\Omega_{G^q, in}$ 、 $\Omega_{F^p, in}$ 、 $\Omega_{H^s, in}$ と出力空間の有限集合

10 $\Omega_{G^q, out}$ 、 $\Omega_{F^p, out}$ 、 $\Omega_{H^s, out}$ の要素数 $|\Omega_{G^q, in}|$ 、 $|\Omega_{F^p, in}|$ 、 $|\Omega_{H^s, in}|$ 、 $|\Omega_{G^q, out}|$ 、 $|\Omega_{F^p, out}|$ 、 $|\Omega_{H^s, out}|$ に対して、これらの最大要素数に対し

$$\max\{|\Omega_{G^q, in}|, |\Omega_{F^p, in}|, |\Omega_{H^s, in}|, |\Omega_{G^q, out}|, |\Omega_{F^p, out}|, |\Omega_{H^s, out}|\} \leq r^n$$

なる基数 r 、語長 n の数表現の集合 B_r^n を考え、これを旧演算系の旧表現データの集合として扱い、前記の元演算系の各演算(元演算)の入力空間のデータ集合

15 Ω_{in} ($\Omega_{G^q, in}$ 、 $\Omega_{F^p, in}$ 、 $\Omega_{H^s, in}$ の何れか)の要素数 $|\Omega_{in}|$ に対して、 $|\Omega_{in}| = r^n$ なる時以外は、 $r^n - |\Omega_{in}|$ 個分の定まっていない要素に対する対応関係をそれぞれの元演算毎に付加し、元演算の全てを B_r^n 上の演算になるように拡張する。

前記元演算系が Q 種の単項演算を含む場合には、これらを拡張し

$G^q_o: B_r^n \rightarrow B_r^n$ ($q=1,2,\dots,Q$)なる単項演算として扱い、また P 種の2項演算を

20 含む場合には、これらを拡張し $F^p_o: B_r^n \times B_r^n \rightarrow B_r^n$ ($p=1,2,\dots,P$)なる2項演算と

して扱い、また S 種の T 項演算を含む場合には、これらを拡張し

$H^s_o: B_r^n \times B_r^n \times \dots \times B_r^n \rightarrow B_r^n$ (B_r^n の T 個の直積、 $s=1,2,\dots,S$)なる T 項演算として扱う。以後 B_r^n 、 G^q_o 、 F^p_o 、 H^s_o を合わせて旧演算系と呼ぶことにする。

さらに集合 B_r^m ($m \geq n$)上のデータを新表現データとして扱い、符号器は

25 $\Phi: B_r^n \rightarrow B_r^m$ なる単射の写像として扱い、復号器は $\Psi: B_r^m \rightarrow B_r^n$ なる全射の写像として扱い、さらに G^q_o に対応する新演算系の演算は $G^q_N: B_r^m \rightarrow B_r^m$ なる単項

演算として扱い、 F^p_o に対応する新演算系の演算は $F^p_N: B_r^m \times B_r^m \rightarrow B_r^m$ なる二

項演算として扱い、 H^s_o に対応する新演算系の演算は

$H^s_N: B_r^m \times B_r^m \times \cdots \times B_r^m \rightarrow B_r^m$ なる T 項演算として扱うことで、旧演算系の全ての演算と新演算系の全ての演算および符号器、復号器を全て多入力多出力の r 値

- 5 論理関数（ r 値論理回路）に対応させることができる。そして関連特許文献に対応した

$$(1) \Phi(X) \in [X] \subset B_r^m \quad (\text{f o r } \forall X \in B_r^n)$$

$$(2) \Psi([X]) = X \quad (\text{f o r } \forall X \in B_r^n)$$

$$(3) Y = G^q_o(X) \Leftrightarrow [Y] \supset G^q_N([X]) \quad (\text{f o r } \forall X, Y \in B_r^n, \forall q)$$

$$10 \quad (4) Z = F^p_o(X, Y) \Leftrightarrow [Z] \supset F^p_N([X], [Y]) \quad (\text{f o r } \forall X, Y, Z \in B_r^n, \forall p)$$

$$(5) Y = H^s_o(X_1, \dots, X_T) \Leftrightarrow [Y] \supset H^s_N([X_1], \dots, [X_T])$$

$$(\text{f o r } \forall X_1, \dots, X_T, Y \in B_r^n, \forall s)$$

なる符号化条件（符号化条件の意味は関連特許文献と同じ。）を課すことで新演算系を具体的な r 論理関数（論理回路）として論理設計を可能とすることができる。

- 15 ここに $[X]$ は B_r^n 上の任意 X に対応した B_r^m 上の部分集合で、集合族 $\{[X]\}_{X \in B_r^n}$

は関連特許文献と同様に符号と呼び、 $C = \bigcup_{X \in \Omega} [X]$ は符号の全体集合、 $N_c = B_r^m - C$

は非符号の集合である。図 1 は旧表現データの集合 B_r^n の要素 X は符号器

$\Phi: B_r^n \rightarrow B_r^m$ により新表現データの集合 B_r^m 上のデータ $\Phi(X)$ に変換される。また

X に対応した符号 $[X]$ は復号器 $\Psi: B_r^m \rightarrow B_r^n$ により旧表現データ X に変換さ

- 20 れることを示す本発明の符号化概念図である。また図 2 は元演算系の入出力データの表現空間が B_r^n と一致しない場合に入出力データの表現空間を B_r^n に拡張した場合を一つの元演算系の単項演算 G を旧演算系の G_o に拡張した場合で拡張の概念を示す本発明の元演算系の旧演算系への拡張概念図である。

また図 3 は元演算系から旧演算系に拡張後に、新演算系を決定する方法の 1 つ

- 25 で、本発明の旧演算系新演算系決定フロー例である。図 3 の処理は以下のとおりである。

[ステップ S 1]: 符号を生成する。

[ステップ S 2]: 符号化条件 (1) および (2) を判別する。符号化条件 (1) および (2) を満たさない場合にはステップ S 1 に戻り処理を繰り返す。符号化条件 (1) および (2) を満たす場合にはステップ S 3 へ進む。

5 [ステップ S 3]: 新演算器を生成する。

[ステップ S 4]: 符号化条件 (3) 乃至 (5) を判別する。符号化条件 (3) 乃至 (5) を満たさない場合にはステップ S 3 に戻り処理を繰り返す。符号化条件 (3) 乃至 (5) を満たす場合にはステップ S 5 へ進む。

[ステップ S 5]: 新演算系を評価する。

10 [ステップ S 6]: 新演算器の単純化が十分か判別する。十分であれば処理が終了する。十分でなければステップ S 7 へ進む。

[ステップ S 7]: 符号ないのすべての新演算器を生成したかを判別する。すべては生成していない場合にはステップ S 3 へ戻り、すべてせいせいした場合にはステップ S 1 に戻って処理を繰り返す。

15 図 3 の処理の具体例については実施例で後に示す。

このフロー以外にも後の実施例で示すように幾つかの決定フローが存在することとは言ってもない。

以上のような決定フローにより符号および新演算器が決定され、これを実装して演算処理装置が図 4 に示すように構成される。図 4 では、演算処理装置 100
20 が、演算処理手段 101、符号器 102、復号器 103 および伝送路 104 から構成される。演算処理手段 101 は上述の新演算器の構成を実装したものである。符号器 102、復号器 103 は、上述の符号の符号化・復号を実現するように実装される。伝送路 104 は、演算処理手段 101 と符号器 102 との間のデータ転送、演算処理手段 101 と復号器 103 との間のデータ転送等を行うものであり、LSI の内部バス、外部バス、通信路等で構成することができる。
25

尚、前記 F^p は関連特許文献の A_p を、前記 G^q は関連特許文献の B_q をそれぞれ上記のように拡張したものに相当する。以後 B_r^m , G^q_N , F^p_N , H^s_N , Φ , Ψ を合わせて新演算系と呼ぶことにする。

関連特許文献の新表現データの集合 Ω' が B_r^m の形でなく一般的な集合のままであると、新演算系は前記関連特許文献と同様に一般的な写像による表現で扱わざるを得ないが、本発明では新表現データの集合を B_r^m の形に限定することで、旧演算系の演算器および新演算系を全て r 値の論理回路に対応させることが可能

5 になった。このことにより r 値論理で表される写像としてコンピュータによる探索などの方法により新演算系の決定が可能となる。

一般に新演算系は莫大な種類存在するが、探索などで得られたものを評価し、目的に合致するかを判断し、合致しなければ他の新演算系を次々と探索で生成し、目的に合う新演算系を決定する。得られた新演算系は r 値論理回路に対応している

10 のので通常の論理設計の方法で r 値論理回路の設計が可能になる。

尚、2 値以外の r 値論理については例えば、樋口龍雄、亀山充隆著“多値情報処理：ポストバイナリエレクトロニクス”昭晃社に詳しく述べられている。

ところで (1) 乃至 (5) は集合論的記述 (\supset , \Leftrightarrow なる記号によって表される関係式) である。一般に集合の特徴関数 $\chi_A(X)$ の定義から

15 $\chi_A(X) = 1(\text{for } \forall X \in B) \Leftrightarrow B \subset A$ である。このことから (1) 乃至 (5) を同値な特徴関数 (特性関数) による方程式の形で表すことができる。(1) 乃至 (5) に同値な式を特徴関数 (Characteristic function) による方程式であることを陽に示すために番号に c を付し、それぞれ (1c) 乃至 (5c) とすると以下のようになる。

20 (1c) $\chi'_{[X]}(\Phi(X)) = 1 \text{ (for } \forall X \in B_r^n)$

(2c) $\chi_X(\Psi([X])) = 1 \text{ (for } \forall X \in B_r^n)$

(3c) $\chi'_{[G^q_o(X)]}(G^q_N([X])) = 1 \text{ (for } \forall X \in B_r^n, \forall q)$

(4c) $\chi'_{[F^p_o(X,Y)]}(F^p_N([X],[Y])) = 1 \text{ (for } \forall X, Y \in B_r^n, \forall p)$

(5c) $\chi'_{[H^s_o(X_1, \dots, X_T)]}(H^s_N([X_1], \dots, [X_T])) = 1 \text{ (for } \forall X_1, \dots, X_T \in B_r^n, \forall s)$

25 ここに $\chi_A(X)$ は B_r^n の部分集合 A の特徴関数で、 $X \in B_r^n$ 、 $\chi'_A(X')$ は B_r^m の部分集合 A' の特徴関数で、 $X' \in B_r^m$

この (1 c) 乃至 (5 c) を符号化条件として用いれば、与えられた旧演算系の演算器に対し、新演算系が符号化条件を満足するかどうかを特徴関数が 1 となるかどうかで判定すればよく、コンピュータによる判定において扱い易いメリットがある。

5 上記 (1) 乃至 (5) または (1 c) 乃至 (5 c) を符号化条件の下で既知の元演算系の各演算 (元演算) の情報から前記符号化条件を満足する新演算系を r 値論理回路に対応した論理関数として設計することが可能になり、論理設計された新演算系を用いた演算結果が元演算系による演算結果と同じになることを保証できることを特徴とする符号化演算方式が実現できる。

10 さらに $r = 2$ として設計する場合には、特徴関数の値が 0, 1 の 2 値をとり、 $X \in B^n$ であることから、特徴関数 $\chi_S(X)$ を 1 つの n 変数の 2 値論理関数のとみなして扱うことができる。この点に着目すると (1 c) 乃至 (5 c) を、後述の発明の第 3 の側面等に関連して示すような同値な 2 値論理関数による明解な方程式の形で表すことができる。

15

[発明の第 2 の側面]

発明の第 2 の側面は請求項 5 および 6 の発明に対応する。

(6) B^n 上の集合の特徴関数を論理関数として扱う。

(7) B^n 上の集合の 1 つの要素に 1 つの最小項を対応させる。

20 の 2 つを前提の手段とする。

以後特徴関数を 2 値論理関数で表したものを特徴論理関数と呼ぶことにする。

具体的には B^n の部分集合 S ($S \subset B^n$) の特徴論理関数を $\chi_S(X)$ ($X \in B^n$) と表し、また B^m の部分集合 T ($T \subset B^m$) の特徴論理関数を $\chi'_T(Y)$ ($Y \in B^m$) と表すことにし、 B^n の部分集合 S の写像 $F: B^n \rightarrow B^m$ による像を $F(S)$ と表すと、

25 $\chi'_{F(S)}(Y) = 1$ ($Y \in F(S)$ の時)、 $\chi'_{F(S)}(Y) = 0$ ($Y \notin F(S)$ の時) となる。

そこで B^n 上の 1 つの要素 $Q (= (q_1, q_2, \dots, q_n))$ に 1 つの最小項からなる 1 要素集合の特徴論理関数を $\chi_Q(X)$ と表すと、これは $X = Q$ の場合のみ 1 となるから、明らかに $\chi_Q(X) = X^Q$ が成り立つ。最小項間は分離的 ($X^{Q1} \cdot X^{Q2} = 0$ ($Q1 \neq Q2$ の時)) なので、明らかに

$$(8) \quad \chi_S(X) = \bigcup_{Q \in S} \chi_Q(X) = \bigcup_{Q \in S} X^Q$$

と表せる。同様に B^m 上の 1 要素 P を考えると全く同様に、

$$(8-1) \quad \chi'_{F(S)}(Y) = \bigcup_{P \in F(S)} \chi'_P(Y) = \bigcup_{P \in F(S)} Y^P$$

となる。

- 5 P が $F(S)$ 全体を動く場合と、 $P = F(Q)$ と置いて Q が S 全体を動く場合の P の動く範囲は全く同じであることに注意すると、(8-1) 式は

$$(9) \quad \chi'_{F(S)}(Y) = \bigcup_{Q \in S} Y^{F(Q)}$$

となる。

- 10 S に制限された範囲で和をとることと、 $\chi_S(X)$ を掛けて全体で和をとっても特徴関数の性質から明らかに同じである。従って

$$(10) \quad \chi'_{F(S)}(Y) = \bigcup_X Y^{F(X)} \cdot \chi_S(X)$$

が成り立つ。また B^n の部分集合 $S, T \subset B^n$ の特徴論理関数に対して

$$(11) \quad \chi_S(X) \cdot \chi_T(X) = 0 \quad (\text{for } \forall X \in B^n) \Leftrightarrow S \cap T = \emptyset$$

$$(12) \quad \chi_{S \cap T}(X) = \chi_S(X) \cdot \chi_T(X)$$

$$15 \quad (13) \quad \chi_{S \cup T}(X) = \chi_S(X) \cup \chi_T(X)$$

$$(14) \quad \overline{\chi_S(X)} \cdot \chi_T(X) = 0 \quad (\text{for } \forall X \in B^n) \Leftrightarrow S \supset T$$

なる関係式が明らかに成り立つ。

以上のような構成により、集合の特徴関数を論理的に扱うことを可能とし、

(8) 乃至 (14) は B^n 上の一般的な集合の集合計算に用いることができる。

- 20 さらに、 $F: B^n \rightarrow B^m$ 、 $G: B^m \rightarrow B^l$ なる写像を考え、 F, G の合成写像を $R: B^n \rightarrow B^l$ とすると、論理関数をベクトル型で示せば $R(X) = G(F(X))$ なる関係がある。 B^l の部分集合 U ($U \subset B^l$) の特徴論理関数を $\chi''_U(Z)$ ($Z \in B^l$) と表すことにすると、上記 (10) と同様に

$$(10-1) \quad \chi''_{R(S)}(Z) = \bigcup_X Z^{R(X)} \cdot \chi_S(X)$$

$$(10-2) \quad \chi''_{G(T)}(Z) = \bigcup_Y Z^{G(Y)} \cdot \chi'_T(Y)$$

が成り立つ。

ここで $T = F(S)$ とすると、 $G(F(S)) = R(S)$ だから (10)、(10-1) (10-2) より

$$(10-3) \quad \bigcup_X Z^{R(X)} \cdot \chi_S(X) = \bigcup_X \left(\bigcup_Y Z^{G(Y)} \cdot Y^{F(X)} \right) \cdot \chi_S(X)$$

が導け、これが任意の部分集合 S に対して成り立つことから

$$(15) \quad Z^{R(X)} = \bigcup_Y Z^{G(Y)} \cdot Y^{F(X)}$$

が導かれる。

10 ところで写像 $F: B^n \rightarrow B^m$ に対する $Y^{F(X)}$ は Y, X の 2 値論理関数である。従って $Y^{F(X)}$ は 2^m 種の Y を行変数 (行番号) とし、 2^n 種の X を列変数 (列番号) とした $2^m \times 2^n$ のブール値行列 (Boolean Matrix) となる。

そして (15) 式は、写像 F に対応した行列 $Y^{F(X)}$ と写像 G に対応した行列 $Z^{G(Y)}$ の行列として積計算で合成写像 R に対応した行列 $Z^{R(X)}$ が求まることを示している。
15 $Y^{F(X)}$ を行列表現であることを陽に示す為に、これを写像と同じ大文字を用い $\tilde{F}(Y, X)$ (以後これを写像 F の母関数と呼ぶ。) と置くと、

$$(16) \quad \tilde{F}(Y, X) = Y^{F(X)} = \bigcap_{j=1}^m \{y_j \cdot f_j(X) \cup \overline{y_j} \cdot \overline{f_j(X)}\}$$

となる。

そこで母関数の形で (15) 式を再記すると

$$20 \quad (17) \quad \tilde{R}(Z, X) = \bigcup_Y \tilde{G}(Z, Y) \cdot \tilde{F}(Y, X)$$

と表せる。

また Y の任意の関数 $g(Y)$ に関して論理関数の積和標準形と (16) より

$$(18) \bigcup_Y g(Y) \cdot \tilde{F}(Y, X) = g(F(X))$$

が成り立つ。

また恒等変換 $I: B^n \rightarrow B^n$ に対して

$$(19) \bigcup_Y g(Y) \cdot \tilde{I}(Y, X) = g(X)$$

5 これから写像 F の成分関数と母関数の関係

$$(20) f_j(X) = \bigcup_Y y_j \cdot \tilde{F}(Y, X)$$

$$(21) \overline{f_j(X)} = \bigcup_Y \overline{y_j} \cdot \tilde{F}(Y, X)$$

が導かれる。

また $\Omega = B^n$ と置くと

$$10 \quad (22) \bigcup_Y \tilde{F}(Y, X) = \chi_\Omega(X) = 1$$

$$(23) \bigcup_X \tilde{F}(Y, X) = \chi'_{F(\Omega)}(Y)$$

となり、(22) は写像 F の定義域の特徴論関数 ($= 1$) を、(23) 写像 F の値域の特徴論関数を示している。(17) 乃至 (23) から解るように、母関数は写像の基本的な性質を簡単な演算により導くことができる基本的に重要な表現
15 であることが解る。

さらに、 $\Omega = B^n$ 上の同型写像 $\Phi: \Omega \rightarrow \Omega$ に対して、逆写像を $\Phi^{-1}: \Omega \rightarrow \Omega$ とすると

$$(24) \tilde{\Phi}^{-1}(X, Y) = \tilde{\Phi}(Y, X) \quad (\text{for } \forall X, Y \in \Omega)$$

が成り立つ。これと (20) 式を用いると

$$20 \quad (25) \phi^{-1}_i(X) = \bigcup_Y y_i \cdot \tilde{\Phi}(X, Y)$$

が導かれる。これは逆写像の成分関数を同型写像の母関数から直接求める式になっている。

このことは同型写像は全単射で 1 対 1 対応であることから明らかである。

従って、 $\Omega = B^n$ 上の恒等変換を $I : \Omega \rightarrow \Omega$ とする、 $\Phi^{-1}\Phi = \Phi\Phi^{-1} = I$ に対応し、

$$(26) \quad \bigcup_Z \tilde{\Phi}^{-1}(X, Z) \cdot \tilde{\Phi}(Z, Y) = \bigcup_Z \tilde{\Phi}(X, Z) \cdot \tilde{\Phi}(Z, Y) = \tilde{I}(X, Y) \quad (\text{for } \forall X, Y, Z \in \Omega)$$

が成り立つ。

5 また写像 $F : B^n \rightarrow B^m$ の母関数に対し

$$(27) \quad \tilde{F}(Y, X) \cdot \tilde{F}(Z, X) = \tilde{F}(Y, X) \cdot \tilde{I}(Y, Z) \quad (\text{ここに } Y, Z \in B^m)$$

が成り立つ。

これは (16) より

$$\begin{aligned} \tilde{F}(Y, X) \cdot \tilde{F}(Z, X) &= \bigcap_{j=1}^m (y_j f_j(X) \cup \overline{y_j} \cdot \overline{f_j(X)}) \cdot (z_j f_j(X) \cup \overline{z_j} \cdot \overline{f_j(X)}) \\ &= \bigcap_{j=1}^m (y_j z_j f_j(X) \cup \overline{y_j} \cdot \overline{z_j} \cdot \overline{f_j(X)}) = \bigcap_j (y_j f_j(X) \cup \overline{y_j} \cdot \overline{f_j(X)}) \cdot (y_j z_j \cup \overline{y_j} \cdot \overline{z_j}) \\ &= \tilde{F}(Y, X) \cdot \tilde{I}(Y, Z) \end{aligned}$$

10 と導かれる。

ところで $X \in B^n$ 、 $Y \in B^m$ のある関数 $\alpha(Y, X)$ がある写像 $B^n \rightarrow B^m$ の母関数である為の必要十分条件は明らかに “ $\alpha(Y, X)$ を各 X ごとの列ベクトルが唯一の Y に対して $\alpha(Y, X) = 1$ となる。” という命題になる。

“1 つの関数 $g(Y)$ が唯一の Y に対して $g(Y) = 1$ となる。” という命題は

$$15 \quad (28) \quad \bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot g(Y) = \overline{g(Z)} \quad (\text{ここに } Y, Z \in B^m)$$

が成り立つことと同値である。先ずこれを証明すると、

$$a) \quad g(Y) = 0 \text{ の時、} \quad \bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot g(Y) = 0 \neq \overline{g(Z)} = 1$$

b) $g(Y)$ が 2^m 個の Y の中の 1 ヶ所 P で 1 その他で 0 の時、 $\overline{\tilde{I}(Z, P)}$ は $Z = P$ で 0 でそれ以外 1 であるので (28) が成り立つ。

$$20 \quad c) \quad g(Y) \text{ が } 2^m \text{ 個の } Y \text{ の中の 2 ヶ所で 1 の場合 } \bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot g(Y) = 1 \neq \overline{g(Z)}$$

a), b), c) で全ての場合を尽くすことから (28) が証明された。このよ

うな関数 $g(Y)$ は 1 つの最小項からなるので最小項関数と呼ぶことにする。

このことから前記の $\alpha(Y, X)$ がある写像の母関数である為の必要十分条件は

$$(29) \bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot \tilde{F}(Y, X) = \overline{\tilde{F}(Z, X)} \quad (\text{for } \forall Y, Z \in B^m, \forall X \in B^n)$$

となる。

- 5 以上のように、多入力多出力の写像を成分関数毎に扱うのでは無く、母関数により一括して扱うことを可能にし、また各種の論理関数間の関係を見通しよく効率的に計算することが可能になる特徴を有している。

[発明の第 3 の側面]

- 10 発明の第 3 の側面は請求項 7 の発明に対応する。また、この第 3 の側面を、先の第 2 の側面と合わせたものは、請求項 8 の発明に対応する。

先ず論理関数 $f(X)$ に対して

$$(30) f(X) \cdot \overline{f(Y)} \cdot \Delta(X, Y) = 0$$

- 15 が成り立つかどうかを各種の $\Delta(X, Y)$ に対して判定することで、論理関数 $f(X)$ の複雑さを判定することを可能にし、また写像 $F: B^n \rightarrow B^m$ の各成分関数 $f_j(X)$ ($j=1, 2, \dots, m$) に対して写像の母関数を用い

$$(31) \tilde{F}(X, A) \cdot \tilde{F}(Y, B) \cdot x_j \cdot \overline{y_j} \cdot \Delta_j(A, B) = 0$$

が成り立つかどうかを $\Delta_j(A, B)$ に対して判定することで、各論理関数 $f_j(X)$ の複雑さを一括して判定することを可能にすることである。

- 20 「記号と記法」の項で述べたように、関数 $f(X)$ ($X \in B^n$) が変数 x_i に依存し

ていないための必要十分条件は $\frac{df(X)}{dx_i} = 0$ であるが、これは恒等変換の母関数

$\tilde{I}(X, Y)$ を用い

$$(32) f(X) \cdot \overline{f(Y)} \cdot \tilde{I}(X, Y^i) = 0$$

と表すことができる。ここに X^i はベクトル X の i 成分 x_i を $\overline{x_i}$ で置き換えたベクトル

- 25 ルである。

これは (32) の左側 = 0 の式を Y で和をとると、(19) から $f(X) \cdot \overline{f(X^i)} = 0$ となり、またこの式の X に X^i を代入すると $f(X^i) \cdot \overline{f(X)} = 0$ となり、ブール微分

の $\frac{df(X)}{dx_i} = 0$ と同値であることから解る。(32) は (30) における $\Delta(X, Y)$ を

$\Delta(X, Y) = \tilde{I}(X, Y^i)$ とした場合に相当する。

さらに $f(X)$ が変数 x_i と変数 x_j に依存していない時、

$$f(X) \cdot \overline{f(Y)} \cdot \tilde{I}(X, Y^i) = 0, \quad f(X) \cdot \overline{f(Y)} \cdot \tilde{I}(X, Y^j) = 0, \quad f(X) \cdot \overline{f(Y)} \cdot \tilde{I}(X, Y^{i,j}) = 0$$

- 5 が必要十分条件となる。ここに $X^{i,j}$ はベクトル X の x_i, x_j をそれぞれ $\overline{x_i}, \overline{x_j}$ に置き換えたベクトル。

さらに一般に n 変数の集合の部分集合を Θ で表し、 Θ の部分集合を $L (L \subset \Theta)$ とし、 X^L をベクトル X の変数の内の集合 L に含まれる変数を否定に置き換えたベクトルとした時、集合 Θ の変数に依存しないことの必要十分条件は

$$10 \quad (33) \quad f(X) \cdot \overline{f(Y)} \cdot \tilde{I}(X, Y^L) = 0 \quad (\text{for } \forall L \subset \Theta)$$

と表せる。ここで Θ を n 個のブール値 θ^l により $\Theta = \{l; \theta^l = 0\}$ と表すと、(33) は

$$(34) \quad f(X) \cdot \overline{f(Y)} \cdot \bigcap_l (x_l \cdot y_l \cup \overline{x_l} \cdot \overline{y_l} \cup \overline{\theta^l}) = 0$$

と表すことができる。これは

$$15 \quad \bigcup_{L \subset \Theta} \tilde{I}(X, Y^L) = \bigcap_l (x_l \cdot y_l \cup \overline{x_l} \cdot \overline{y_l} \cup \overline{\theta^l})$$

が成り立つことから導かれる。この θ^l を関数 $f(X)$ の変数依存度と呼ぶことにする。

尚、(34) が成り立つ時、 $f(X)$ は $\theta^l = 0$ なる変数 x_l には依存していない。依存している変数の数は $\theta^l = 1$ なる変数の数以下である。(34) は (30) におけ

$$20 \quad \Delta(X, Y) \text{ を } \Delta(X, Y) = \bigcap_l (x_l \cdot y_l \cup \overline{x_l} \cdot \overline{y_l} \cup \overline{\theta^l}) \text{ とした場合に相当する。}$$

さらに写像 $F: B^n \rightarrow B^m$ の各成分関数 $f_j(X)$ ($j=1, 2, \dots, m$) の変数依存度を θ_j^l とすると、(34) に相当した式を母関数を用い写像全体を一括して

$$(35) \quad \tilde{F}(X,A) \cdot \tilde{F}(Y,B) \cdot x_j \cdot \overline{y_j} \cdot \bigcap_i (a_i \cdot b_i \cup \overline{a_i} \cdot \overline{b_i} \cup \theta_j') = 0$$

と表すことができる。

この(35)は(31)の $\Delta_j(A,B)$ を $\Delta_j(A,B) = \bigcap_i (a_i \cdot b_i \cup \overline{a_i} \cdot \overline{b_i} \cup \theta_j')$ とした

場合に相当する。この $\Delta(X,Y)$ や $\Delta_j(A,B)$ を今後変数依存度関数と呼ぶ。

- 5 以上の本発明の第3の側面によれば、論理関数の変数依存度を方程式を満足するかどうかで複数の変数に対して一度に判定でき、また上述のこの発明の第2の側面をさらに総合的に適用して論理関数設計を行える。

[発明の第4の側面]

- 10 発明の第4の側面は請求項9の発明に対応する。

$r = 2$ に限定すると、旧演算系の演算（写像）は、 $G^q_o: B^n \rightarrow B^n$ 、

$F^p_o: B^n \times B^n \rightarrow B^n$ 、 $H^s_o: B^n \times B^n \times \cdots \times B^n \rightarrow B^n$ となり、新演算系の符号器（写像）は $\Phi: B^n \rightarrow B^m$ 、復号器（写像）は、 $\Psi: B^m \rightarrow B^n$ 、新演算系の演算（写像）

は $G^q_N: B^m \rightarrow B^m$ 、 $F^p_N: B^m \times B^m \rightarrow B^m$ 、 $H^s_N: B^m \times B^m \times \cdots \times B^m \rightarrow B^m$ となる。

- 15 この時、旧演算系の各演算および新演算系の各演算、符号器、復号器の成分関数は、

$i = 1, 2, \dots, n$ 、 $X, Y, Z, X_1, X_2, \dots, X_T \in B^n$ 、 $j = 1, 2, \dots, m$ 、

$X', Y', Z', X'_1, X'_2, \dots, X'_T \in B^m$ とすると、

$G^q_o: g^q_o(X)$ 、 $F^p_o: f^p_o(X, Y)$ 、 $H^s_o: h^s_o(X_1, X_2, \dots, X_T)$ および $G^q_N: g^q_N(X')$ 、

- 20 $F^p_N: f^p_N(X', Y')$ 、 $H^s_N: h^s_N(X'_1, X'_2, \dots, X'_T)$ 、 $\Phi: \phi_j(X)$ 、 $\Psi: \psi_i(X')$

と表すことができる。これらの写像を上述の発明の第2の側面で定義した母関数を用い、さらに符号領域 C の特徴論理関数 $\chi_c(X')$ を $c(X')$ と表すと、 $r = 2$ においては(1-c)乃至(5-c)と同値な以下の(1-b)乃至(5-b-2)が導かれる。この(1-b)乃至(5-b-2)は2値ブール関数(B o o l e

a n f u n c t i o n) による方程式であることを陽に示すために番号に b を付した。

$$(1-b) \quad \tilde{\Psi}(X, X') \cdot c(X') \cdot \tilde{\Phi}(X', X) = \tilde{\Phi}(X', X)$$

$$(2-b) \quad \bigcup_{X'} \tilde{\Phi}(X', X) \cdot \tilde{\Phi}(X', Y) = \tilde{I}(X, Y)$$

$$5 \quad (3-b-1) \quad \overline{\tilde{G}^q_o(Y, X)} \cdot \tilde{G}^q_N(Y', X') \cdot \tilde{\Psi}(Y, Y')c(Y') \cdot \tilde{\Psi}(X, X')c(X') = 0$$

$$(3-b-2) \quad \overline{c(Y')} \cdot c(X') \cdot \tilde{G}^q_N(Y', X') = 0$$

$$(4-b-1)$$

$$\overline{\tilde{F}^p_o(Z, Y, X)} \cdot \tilde{F}^p_N(Z', Y', X') \cdot \tilde{\Psi}(Z, Z')c(Z') \cdot \tilde{\Psi}(Y, Y')c(Y') \cdot \tilde{\Psi}(X, X')c(X') = 0$$

$$(4-b-2) \quad \overline{c(Z')} \cdot c(Y') \cdot c(X') \cdot \tilde{F}^p_N(Z', Y', X') = 0$$

$$10 \quad (5-b-1)$$

$$\overline{\tilde{H}^s_o(Y, X_1, \dots, X_T)} \cdot \tilde{H}^s_N(Y', X'_1, \dots, X'_T) \cdot \tilde{\Psi}(Y, Y')c(Y') \cdot \tilde{\Psi}(X_1, X'_1)c(X'_1) \cdots \tilde{\Psi}(X_T, X'_T)c(X'_T) = 0$$

$$(5-b-2) \quad \overline{c(Y')} \cdot c(X'_1) \cdots c(X') \cdot \tilde{H}^s_N(Y', X'_1, \dots, X'_T) = 0$$

$\tilde{\Phi}(X', X)$ は符号器の母関数、 $\tilde{\Psi}(X, X')$ は復号器の母関数、 $\tilde{G}^q_o(Y, X)$ は旧演算系の単項演算の母関数、 $\tilde{F}^p_o(Z, Y, X)$ は旧演算系の二項演算の母関数、

15 $\tilde{G}^q_N(Y', X')$ は新演算系の単項演算の母関数、 $\tilde{F}^p_N(Z', Y', X')$ は新演算系の二項演算の母関数、 $\tilde{H}^s_o(Y, X_1, \dots, X_T)$ は旧演算系の T 項演算の母関数、

$\tilde{H}^s_N(Y', X'_1, \dots, X'_T)$ は新演算系の T 項演算の母関数である。

以下 (1-c) 乃至 (5-c) から同値な変形により (1-b) 乃至 (5-b-2) を導く。

20 先ず各 X の符号を表す集合 $[X]$ の特徴関数 $\chi'_{[X]}(X')$ は、復号器 Ψ の母関数を用いて明らかに以下のように表すことができる。

$$(36) \quad \chi'_{[X]}(X') = \tilde{\Psi}(X, X') \cdot c(X')$$

この (36) を X' の関数と考え母関数の性質 (18) を適用すると、(1c) は

$$(37) \quad \bigcup_{X'} \tilde{\Psi}(X, X') \cdot c(X') \cdot \tilde{\Phi}(X', X) = 1$$

25 と同値になる。また母関数の性質 (27) から

$$(38) \quad \tilde{\Phi}(X', X) \cdot \tilde{\Phi}(Y', X) = \tilde{\Phi}(X', X) \cdot \tilde{I}(X', Y')$$

が成り立つので、(37)の両辺に $\tilde{\Phi}(Y', X)$ を掛け、恒等変換の母関数の性質(19)を適用して和をとり、その後に変数 Y' を X' に変更することにより、前記(1-b)式が導かれる。

次に特徴論理関数と母関数の性質(10)および(16)と(2-c)から

$$5 \quad (39) \quad \chi_{\Psi([Y])}(X) = \bigcup_{X'} \tilde{\Psi}(X, X') \cdot \phi'_{[Y]}(X') = \bigcup_{X'} \tilde{\Psi}(X, X') \cdot \tilde{\Psi}(Y, X') \cdot c(X')$$

ところで(2)に戻ると分かるように(2-c)は

$$(40) \quad \chi_{\Psi([Y])}(X) = \chi_Y(X)$$

と同値である。(39)、(40)と母関数の性質(27)より、(2-c)は

$$(41) \quad \bigcup_{X'} \tilde{\Psi}(X, X') \cdot \tilde{I}(X, Y) \cdot c(X') = \tilde{I}(X, Y)$$

10 同値になり、この両辺を Y で和をとると、母関数の性質(22)より(2-b)式が導かれる。

次に特徴関数の定義と性質(14)より(3-c)は明らかに

$$(42) \quad \chi'_{G^q_N([X])}(Y') \cdot \overline{\chi'_{[G^q_o(X)]}(Y')} = 0$$

と同値になる。また(36)より

$$15 \quad (43) \quad \chi'_{[G^q_o(X)]}(Y') = \tilde{\Psi}(G^q_o(X), Y') \cdot c(Y')$$

また(36)と特徴関数と母関数の性質(10)、(16)より

$$(44) \quad \chi'_{G^q_N([X])}(Y') = \bigcup_{X'} \tilde{G}^q_N(Y', X') \cdot \chi'_{[X]}(X') = \bigcup_{X'} \tilde{G}^q_N(Y', X') \cdot \tilde{\Psi}(X, X') \cdot c(X')$$

(43)と(44)を(42)に代入し、=0型のブール式は和を外せるので

$$(45) \quad \tilde{G}^q_N(Y', X') \cdot \tilde{\Psi}(X, X') \cdot c(X') \cdot \overline{\tilde{\Psi}(G^q_o(X), Y') \cdot c(Y')} = 0$$

20 が得られる。この否定部分を分けると

$$(46) \quad \tilde{G}^q_N(Y', X') \cdot \tilde{\Psi}(X, X') \cdot c(X') \cdot \overline{\tilde{\Psi}(G^q_o(X), Y')} = 0$$

$$(47) \quad \tilde{G}^q_N(Y', X') \cdot \tilde{\Psi}(X, X') \cdot c(X') \cdot \overline{c(Y')} = 0$$

なる2つの方程式になる。(46)に母関数の性質(18)と(29)を適用すると

(3-b-1) が導かれ、(47) を X で和をとると母関数の性質 (22) より (3-b-2) が導かれる。全く同じ論法で、(4-c) から (4-b-1) と (4-b-2) が導け、(5-c) から (5-b-1) と (5-b-2) が導かれる。

以上のように (1-b) 乃至 (5-b-2) が導かれ、旧演算系と新演算系の

- 5 符号化条件式を 2 値論理の明解な方程式の形で表すことができた。このことにより、旧演算系の成分関数の情報から新演算系を 2 値の探索などにより効率よく決定することが可能になる。

[発明の第 5 の側面]

- 10 発明の第 5 の側面は請求項 10 ~ 12 の発明に対応する。

前記 (1-b) 乃至 (5-b-2) に加え、新演算系の演算器に対する論理式の簡単化の条件を付加し、その下で符号器、復号器、新演算器を論理設計することで、新演算器の回路規模および演算遅延時間を小さくすることを可能とする。さらに前記演算器の簡単化の条件として単項演算に対しては

$$15 \quad (48) \quad \tilde{G}^q_N(X', A') \cdot \tilde{G}^q_N(Y', B') \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\lambda^{q_j i}}) = 0$$

なる条件を課し、二項演算に対しては

$$(49) \quad \tilde{F}^p_N(X', A', C') \cdot \tilde{F}^p_N(Y', B', C') \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\theta_1^{p_j i}}) = 0$$

$$\tilde{F}^p_N(X', C', A') \cdot \tilde{F}^p_N(Y', C', B') \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\theta_2^{p_j i}}) = 0$$

なる条件を課し、T 項演算に対しては

$$20 \quad (50)$$

$$\tilde{H}^p_N(X', A', C'_2, \dots, C'_T) \cdot \tilde{H}^p_N(Y', B', C'_2, \dots, C'_T) \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\mu_1^{p_j i}}) = 0$$

$$\tilde{H}^p_N(X', C'_1, A', \dots, C'_T) \cdot \tilde{H}^p_N(Y', C'_1, B', \dots, C'_T) \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\mu_2^{p_j i}}) = 0$$

.....

$$\tilde{H}^p_N(X', C'_1, \dots, C'_{T-1}, A') \cdot \tilde{H}^p_N(Y', C'_1, \dots, C'_{T-1}, B') \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\mu_T^s{}^p{}_{j'}}) = 0$$

なる条件を課し、新演算器の入力変数に対する変数依存度を旧演算系に比較し小さくなるように $\lambda^{q'}_{j'}$ および $\theta_1^{p'}_{j'}$ 、 $\theta_2^{p'}_{j'}$ および $\mu_1^{s'}_{j'}$ 乃至 $\mu_T^{s'}_{j'}$ の値を定め、符号器、復号器、新演算器を論理設計することで、新演算器の回路規模および演算遅延時間

5 間を小さくすることを可能とする。

さらに必要があれば前記簡単化の条件において前記 $\theta_1^{p'}_{j'}$ と $\theta_2^{p'}_{j'}$ に対して $\theta_1^{p'}_{j'} = \theta_2^{p'}_{j'}$ なる条件を課し、新演算系の二項演算を対称型で論理設計する。また前記 $\mu_1^{s'}_{j'}$ 乃至 $\mu_T^{s'}_{j'}$ に対して $\mu_1^{s'}_{j'} = \dots = \mu_T^{s'}_{j'}$ なる条件を課し、新演算系の T 項演算を対称型で設計することを可能とする。

10

〔発明の第 6 の側面〕

発明の第 6 の側面は請求項 1 3 および 1 4 の発明に対応する。

前記旧表現データの空間 B_r^n と新表現データの空間 B_r^m を同一の空間

($B_r^n = B_r^m, n = m$) とした時、符号器 Φ を $\Phi : B_r^n \rightarrow B_r^n$ なる同型写像とし、復
15 号器 Ψ を $\Psi = \Phi^{-1}$ (Φ の逆写像) とすることが考えられる。この場合、符号器が同型写像であれば、前記の (1) および (2) は自動的に満足されることは明らかである。

ここで $r = 2$ の時には (1 - b) および (2 - b) の代わりに $\tilde{\Phi}(Y, X)$ が同型写像の母関数である条件

$$\begin{aligned} & \bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot \tilde{\Phi}(Y, X) = \overline{\tilde{\Phi}(Z, X)} \\ 20 \quad (51) \quad & \bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot \tilde{\Phi}(X, Y) = \overline{\tilde{\Phi}(X, Z)} \quad (\text{for } \forall X, Y, Z \in B^n) \end{aligned}$$

が成り立てばよいことになる。尚、(51) は (29) を入力出力を入れ替えて適用すれば導かれる。また単項演算の (3 - b - 1) (3 - b - 2) に相当する新旧演算系間の関係式は

$$(52) \quad \tilde{G}^q_N(Y', X') = \bigcup_Y \bigcup_X \tilde{G}^q_o(Y, X) \cdot \tilde{\Phi}(Y', Y) \cdot \tilde{\Phi}(X', X)$$

25 が成り立つことに置き換わる。同様に

$$(53) \quad \tilde{F}^p_N(Z', X', Y') = \bigcup_Z \bigcup_X \bigcup_Y \tilde{F}^p_o(Z, X, Y) \cdot \tilde{\Phi}(Z', Z) \cdot \tilde{\Phi}(X', X) \cdot \tilde{\Phi}(Y', Y)$$

$$(54) \quad \tilde{H}^s_N(Y', X'_1, \dots, X'_T) = \bigcup_Y \bigcup_{X_1} \dots \bigcup_{X_T} \tilde{H}^s_o(Y, X_1, \dots, X_T) \cdot \tilde{\Phi}(Y', Y) \cdot \tilde{\Phi}(X'_1, X_1) \dots \tilde{\Phi}(X'_T, X_T)$$

この時、若し新演算系に対して (48) の変数依存度を要請するならば、(52) を (48) に適用し、 X', Y', A', B' に関して和をとると、

5 (55)

$$\tilde{G}^q_o(X, A) \cdot \tilde{G}^q_o(Y, B) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_i (\phi_i(A) \cdot \phi_i(B) \cup \overline{\phi_i(A)} \cdot \overline{\phi_i(B)} \cup \overline{\lambda^q_{j^i}}) = 0$$

なる式が導かれる。

同様に (49) (50) の変数依存度を要請すれば

(56)

$$10 \quad \tilde{F}^p_o(X, A, C) \cdot \tilde{F}^p_o(Y, B, C) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_i (\phi_i(A) \cdot \phi_i(B) \cup \overline{\phi_i(A)} \cdot \overline{\phi_i(B)} \cup \overline{\theta^p_{1^i}}) = 0$$

$$\tilde{F}^p_o(X, C, A) \cdot \tilde{F}^p_o(Y, C, B) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_i (\phi_i(A) \cdot \phi_i(B) \cup \overline{\phi_i(A)} \cdot \overline{\phi_i(B)} \cup \overline{\theta^p_{2^i}}) = 0$$

(57)

$$\tilde{H}^s_o(X, A, C_2, \dots, C_T) \cdot \tilde{H}^s_o(Y, B, C_2, \dots, C_T) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_i (\phi_i(A) \cdot \phi_i(B) \cup \overline{\phi_i(A)} \cdot \overline{\phi_i(B)} \cup \overline{\mu^s_{1^i}}) = 0$$

$$\tilde{H}^s_o(X, C_1, A, \dots, C_T) \cdot \tilde{H}^s_o(Y, C_1, B, \dots, C_T) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_i (\phi_i(A) \cdot \phi_i(B) \cup \overline{\phi_i(A)} \cdot \overline{\phi_i(B)} \cup \overline{\mu^s_{2^i}}) = 0$$

15

.....

$$\tilde{H}^s_o(X, C_1, C_2, \dots, A) \cdot \tilde{H}^s_o(Y, C_1, C_2, \dots, B) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_i (\phi_i(A) \cdot \phi_i(B) \cup \overline{\phi_i(A)} \cdot \overline{\phi_i(B)} \cup \overline{\mu^s_{T^i}}) = 0$$

が導かれる。(55)、(56)、(57) は旧演算系の演算器の母関数と符号成分関数だけの方程式であることに注目すると、次のような決定手順が可能になる。

<ステップ1> 与えられた変数依存度に対して、(51) と (55) 乃至 (57)

20 を満足する符号器 Φ (の成分関数) を決定する。

<ステップ2> (52) 乃至 (54) より新演算系の演算器を決定する。

<ステップ3>復号器 Ψ は $\Psi = \Phi^{-1}$ 、母関数で言えば $\tilde{\Psi}(Y, X) = \tilde{\Phi}(X, Y)$ として決定する。

(ステップ2、ステップ3は明らかに逆の順でも可能。)

この決定手順ではステップ1には新演算系の演算器の情報が含まれていないため、決定すべきファクターが少なく、効率的な符号決定が可能になる。

本発明では以上のような手法により論理設計を行うことで、冗長符号も含めた符号化演算方式の具体的な論理設計を可能とし、さらに上記のように符号化演算方式に基づき論理設計された論理式から具体的な回路設計を行い、符号化演算方式の流れに従い、計算結果を算出できるようにしたこと特徴とする符号化演算装置を実現する。

また1つまたは複数種の新演算器を有したこの符号化演算装置において、これらの新演算器を直接接続することで新表現データに対する複数の計算処理に対し、メモリー経由による遅延時間を無くし、高速な符号化演算装置を実現する。

さらにCPUまたはデータ制御装置の制御の下に、1つまたは複数の新演算器による新表現データとして得られた演算結果を復号器により旧表現データに直ぐに変換するのではなく、新表現データのままで一旦メモリーに格納し、他の計算プロセスにおいて計算された新表現データを前記CPUまたはデータ制御装置により前記結果の新表現データとの新演算器による演算を可能にし、プログラムに従った柔軟な計算処理などを変更可能にした符号化演算装置を実現する。以上のような手段により本発明では前記の関連特許文献の問題点のvi)を解決する。

なお、この発明は装置またはシステムとして実現できるのみでなく、方法としても実現可能である。また、そのような発明の一部をソフトウェアとして構成することができることはもちろんである。またそのようなソフトウェアをコンピュータに実行させるために用いるソフトウェア製品もこの発明の技術的な範囲に含まれることも当然である。

本発明を用いれば、関連特許文献の問題点を解決でき、符号化演算方式に基づく新演算系の具体的な論理設計が具体的且つ効率的に可能となり、元演算系をそのまま回路化した場合に比較し素子数や遅延時間を大幅に削減でき、高速化や低消費電力化を図った符号化演算装置を実現できる。

図面の簡単な説明

図 1 は、本発明の符号化概念図である。

図 2 は、本発明の元演算系の旧演算系への拡張概念図である。

5 図 3 は、本発明の旧演算系新演算系決定フローの例を示す図である。

図 4 は、本発明により構成されて演算処理装置の構成を示す図である。

図 5 は、本発明の旧演算系新演算系決定フローの詳細を示す図である。

図 6 は、本発明の旧演算系新演算系決定フローの他の例の詳細を示す図である。

10 図 7 は、本発明の具体例 1 に関連して元演算系の擬似乗算器 F^1 の真理値表を説明する図である。

図 8 は、本発明の具体例 1 に関連して元演算系の擬似加算器 F^2 の真理値表を示す図である。

図 9 は、本発明の具体例 1 に関連して元演算系の擬似乗算器 F^1 の回路図を示す図である。

15 図 10 は、本発明の具体例 1 に関連して元演算系の擬似加算器 F^2 の回路図を示す図である。

図 11 は、本発明の具体例 1 に関連して新演算系の決定フローを説明する図である。

20 図 12 は、本発明の具体例 1 に関連して新演算系の擬似乗算器 F^1_N の真理値表を説明する図である。

図 13 は、本発明の具体例 1 に関連して新演算系の擬似加算器 F^2_N の真理値表を示す図である。

図 14 は、本発明の具体例 1 に関連して新演算系の符号器 Φ の真理値表を示す図である。

25 図 15 は、本発明の具体例 1 に関連して新演算系の復号器 Ψ の真理値表を示す図である。

図 16 は、本発明の具体例 1 に関連して新演算系の擬似乗算器 F^1_N の回路図を示す図である。

図 17 は、本発明の具体例 1 に関連して新演算系の擬似加算器 F^2_N の回路図を

示す図である。

図 18 は、本発明の具体例 1 に関連して新演算系の符号器 Φ の回路図を示す図である。

5 図 19 は、本発明の具体例 1 に関連して新演算系の復号器 Ψ の回路図を示す図である。

図 20 は、本発明の具体例 1 に関連して素子数遅延表を示す図である。

図 21 は、本発明の具体例 1 に関連して木構造積和演算器のブロック図を示す図である。

図 22 は、本発明の具体例 1 に関連して擬似積和演算器を示す図である。

10 図 23 は、本発明の具体例 2 に関連して元演算系の 2 つの単項演算 G^1 、 G^2 の真理値表を示す図である。

図 24 は、本発明の具体例 2 に関連して元演算系の 2 つの単項演算 G^1 、 G^2 の母関数ブール値行列を示す図である。

図 25 は、本発明の具体例 2 に関連して新演算系決定フローを示す図である。

15 図 26 は、本発明の具体例 2 に関連して元演算系の単項演算 G^1 、 G^2 の入出力トポロジーを示す図である。

図 27 は、本発明の具体例 2 に関連して新演算系の単項演算 G_N^1 、 G_N^2 の入出力トポロジーを示す図である。

20 図 28 は、本発明の具体例 2 に関連して新演算系の単項演算 G_N^1 、 G_N^2 の束ねた入出力トポロジーを示す図である。

図 29 は、本発明の具体例 2 に関連してトポロジー一致型新演算系決定フローである。

発明を実施するための最良の形態

25 以下、この発明の実施例について説明する。

ここでは、請求項 3～8 の方法の発明について言及していくが、請求項 1 および 2 の装置の発明や請求項 9～14 の方法の発明についても適宜適用されることに留意されたい。

[実施例 1]

請求項 3 の発明の実施例について説明する。

5 先ず図 3 のフローを $r = 2$ で $n = 2$ の旧演算として 1 つの単項演算 G_o が、成分関数として $g_{o0}(A) = a_1, g_{o1}(A) = a_1 \cdot a_0$ と与えられ、 $m = 3$ の新演算器を決定する請求項 3 と請求項 4 の符号化条件による実施例を示す。図 5 は請求項 3 の発明の実施例（請求項 4 の発明の実施例も同様）のプログラムのゼネラル・フローである。先ずこのプログラムに用いた変数と配列は以下のとおりである。

x : 旧符号非符号変数 ; X のとる値 $0 \sim 3$ と非符号（値は 4）に対応させた変数

10 y : 新符号変数 ; X' のとる値 $0 \sim 7$ に対応させた変数

r : 乱数変数 ; 生成された 8 ビット乱数の $0 \sim 255$ を表す変数

i, j, k ; 新演算器生成の更新変数

l, t, u, v : 一般の変数

15 $c[4]$: 生成符号配列 ; 4 種の符号 $[X]$ に対応させた 4 個の配列で、生成した符号が属す新符号の値に対応したビットだけ 1 とし、属さないビットは 0 とする配列、新符号の値とビットの対応には L S B 側から順に対応させる。

$g_o[4]$: 旧演算器写像配列 ; $g_o[0] = 0, g_o[1] = 0, g_o[2] = 1, g_o[3] = 3$

20 $g_n[8]$: 新演算器写像配列 ; 各ステップで生成された新演算器の写像入力値を配列アドレスに対応させ、その時の出力を値とした配列

$g_{ne}[3]$: 新演算器成分関数 ; 新演算器の 3 つの成分関数を表す配列で、各成分論理式を最小項 X^0, \dots, X^7 が含まれる場所の対応するビットを 1 にして L S B 側から対応させて表した配列

$g_{nc}[4]$: 新演算器符号代入関数 :

25 $s[6]$: 3 ビット標準成分関数配列 ; 3 ビットの 3 種の 1 変数論理関数（非否定型）とその否定型 1 変数論理関数を、 $g_{ne}[3]$ と同様に表現した配列で表す。

具体的には 1 変数論理関数（非否定型）のビットパターンが $01010101, 00110011, 00001111$ の 3 種であることに対応し、

$s[0] = 170$, $s[1] = 204$, $s[2] = 240$ としその否定を
続く配列に入れる。つまり

$s[3] = 85$, $s[4] = 41$, $s[5] = 15$

$b_0[8]$: 1ビット0パターン配列; $b[t]$ は t ビット目が0その他1の配列

5 $b_1[8]$: 1ビット1パターン配列; $b[t]$ は t ビット目が1その他0の配列

以下図5のフローの請求項3の場合の具体的な処理を説明する。

[ステップS11]: 初期化。

$go[4]$, $s[3]$, $sn[3]$ 以外の配列と変数は0にする。

[ステップS12]: 符号生成。

10 乱数 r を生成し、新符号 y を $\text{mod}(r, 5)$ に対応した符号 $[X]$ (または N
 C) に属すと考え、 $c[\text{mod}(r, 5)]$ の y 番目のビットに1を立てる。

[ステップS13、S14]: $y = y + 1$ とし $y = 8$ ならステップ2、その他はス
テップS15に進む。

[ステップS15]: 符号化条件(1)(2)式の判定。

15 符号化条件(1)満たすことは4つの $[X]$ には少なくとも1つ新符号が属して
いることと等価である。これは $c[0] \sim c[3]$ が全て0で無いことに対応す
る。1つでも0があればステップ1に戻る。無ければステップS16に進む。

例えば $[0] = \{6'\}$, $[1] = \{2'\}$, $[2] = \{0', 4'\}$, $[3] = \{1', 3',$
 $5', 7'\}$, $NC = \text{空集合}$ なる符号生成がなされた場合には $c[0] = 64$, $c[1]$
20 $= 4$, $c[2] = 17$, $c[3] = 85$ なる値になり、ステップ5に進む。

[ステップS16]: 新演算器生成。

符号条件(1)を満足すれば符号器 Φ と復号器 Ψ は存在は保証されているので
ここでは新演算器だけを生成する。この実施例では旧演算器が and ゲート1個
であることから単純化は1変数依存の論理関数を生成すること限定する。従っ
25 て新演算器の成分関数は非否定型標準成分関数と否定型標準成分関数の計6種
の何れかである。従って $gne[0] = s[i]$, $gne[1] = s[j]$, gne
 $[2] = s[k]$ として新演算器を生成する。

[ステップS17]: 符号化条件(3)式の判定。

先ず $gne[3]$ から $gn[8]$ を計算する。これは次のように計算できる。

同様に $g_{ne}[0]$, $g_n[1]$, $g_n[2]$ の下から t 番目のビットの値を $g_n[t]$ の 0 ビット目 (LSB), 1 ビット目、2 ビット目のそれぞれに移し、3 ビット目以上は 0 のままとする。この処理を $t = 0$ から $t = 7$ まで行うことで $g_n[8]$ を決める。

- 5 次に $c[u]$ の 1 が立っているビットの位置の値 t (LSB は 0 で 1 ビット上に行く毎に 1 増える値) 全てに対して v の $g_n[t]$ ビット目を 1 にする。

そして $c[g_o[u]]$ のビットパターンが v のビットパターンを含んでいる時は $u = u + 1$ とし上記処理を $u = 3$ まで繰り返す、途中で含んでいない場合があれば

- 10 符号化条件 (3) を満足していないことになるからステップ S 1 8 へ進む。最後まで行った場合には符号化条件 (3) は満足されたことになりステップ S 2 0 8 に進む。

[ステップ S 1 8、S 1 9]: i , j , k の値を更新する。

- 15 上記更新後にステップ 5 に進む。但し i , j , k が 0 ~ 5 までの全ての値をとった時には、1 変数依存の新演算器が存在しなかったことに相当するので変数などを初期化後にステップ S 1 2 に戻る。

「ステップ S 2 0」:

- 20 符号化条件の (1) 式に基づき [0] ~ [3] の中の新符号を配列 $c[4]$ のビットパターンの 1 なる部分から 1 つずつ選び、例えば $0 \rightarrow 1'$, $1 \rightarrow 0'$, $2 \rightarrow 4'$, $3 \rightarrow 2'$ なる対応として決める。符合器は一般に幾通りも存在するので論理式が簡単になるものを選択すれする。符号化条件 (2) を満足する復号器は存在することが保証されているのでこの実施例ではここで終了する。

尚、変数などは必要な所で初期化する。

以上が図 5 の処理の内容である。

- 25 この実施例において、ステップ 4 で例として示した符号が乱数により生成され、 $i = 0$, $j = 3$, $k = 1$ の時に終了した。

この結果は符号が $[0] = \{6'\}$, $[1] = \{2'\}$, $[2] = \{0', 4'\}$, $[3] = \{1', 3', 5', 7'\}$, $NC = \text{空集合}$ の時に新演算器として論理式が $g_{N_0}(A') = a'_1, g_{N_1}(A') = a'_2, g_{N_2}(A') = a'_0$ なる解が得られたことを意味する。

尚、この実施例では旧演算器が1つの単項演算だけの場合で示したが、複数の2項演算やT項演算を含む場合にも符号化条件(3)に加え(4)(5)の判定が追加される点とステップ5の新演算器の生成の部分の処理が複雑化する点以外は基本的に同様のフローにより、コンピュータ処理で実現できる。

- 5 以上のように請求項3に基づく演算処理装置設計方法により少なくとも旧演算器よりも同等以下の単純化を達成した新演算器を決定することができる。

[実施例2]

- 次に請求項4の発明の演算処理装置設計方法の実施例を説明する。この実施例
10 においては、図5のフロー中、符号化条件の式(1)乃至(3)を(1c)乃至(3c)に置き換えた処理のよる決定になる。この部分の処理が請求項3の発明の実施例と異なる点はステップS12で生成された符号から以下のように特徴論理関数を計算し判定するステップS17と符号器の決定のステップS20の部分だけである。

- 15 前記と同じ旧演算器の場合で $[0] = \{6'\}$, $[1] = \{2'\}$, $[2] = \{0', 4'\}$, $[3] = \{1', 3', 5', 7'\}$, $NC = \text{空集合}$ のように符号生成され、新演算器も $g_{N0}(A') = a'_0$, $g_{N1}(A') = \overline{a'_0}$, $g_{N2}(A') = a'_1$ と生成した場合で示す。

- 先ずステップ2で生成された符号から $c[4]$ を求めたが実はこれは B^3 上の特徴論理関数である各 X 毎の論理関数 $\chi'_{[X]}(X')$ にそのまま対応している。つまり c
20 $[u]$ の u は X に相当し、各 u 毎の $c[u]$ のビットパターンが $gne[3]$ と同様の表現による B^3 上の論理関数になっている。

そこでステップ6の符号化条件(3)を(3c)による判定に以下のように変更する。(請求項4の発明の実施例のステップS17)

- 請求項3の場合と同様に $gne[3]$ から $gn[8]$ を計算し、 v を同様に計
25 算する。そして $c[go[u]]$ と v のビットパターンの否定との論理和(or)を計算し、計算結果が255かどうかを判定する。これは(3c)式の左辺の特徴論理関数が1であると判定することに相当する。

$u = u + 1$ とし上記処理を $u = 3$ まで繰り返す、途中で255でない場合があれば

符号化条件 (3 c) を満足していないことになるからステップ S 1 8 へ進む。
 最後まで行った場合には符号化条件 (3 c) は満足されたことになりステップ S 2 0 に進む。

(請求項 4 の発明の実施例のステップ S 2 0)

- 5 u の 0 ~ 3 の値に対し、 $c[u]$ と $b_0[t]$ との論理和が 2 5 5 になるかを判定し、符号器を決定する。これは符号化条件 (1 c) による符号器を決定することに相当する。

復号器に対しては請求項 3 のステップ S 2 0 と同様である。

- 10 以上が請求項 4 の発明の実施例の異なる点で、2 つのステップが手続的な処理から簡単な論理演算と等号判定に置き換わり、より簡潔なプログラム処理になる。

このように請求項 4 の発明の演算処理装置設計方法によれば効率的なプログラム処理が可能となり、設計に必要な処理時間の短縮に効果がある。

15 [実施例 3]

次に請求項 5 の発明の実施例を 2 つ示す。

1 つ目は上述の請求項 4 の実施例の 2 つのステップの中で論理和演算を行っている部分は請求項 5 の (1 4) 式 (その否定) の実施例になっている。

- 20 2 つ目また上述のステップ 2 の符号生成に請求項 5 の (8) を適用すれば新符号 y を $\text{mod}(r, 5)$ に対応した符号 $[X]$ (または NC) に属すと考え、 $c[\text{mod}(r, 5)]$ を $c[\text{mod}(r, 5)]$ と $b_1[y]$ との論理和に置き換える。

[実施例 4]

- 25 次に請求項 5 の発明の実施例について説明する。請求項 5 の (9) 乃至 (1 3) 式も同様に集合の要素間の処理に利用できる。従って請求項 5 の論理関数設計方法によれば手続的な処理を簡単な論理演算に置き換えることができ、効率的な処理が可能となる。

[実施例 5]

次に請求項 6 の発明の実施例を以下に示す。

請求項 6 の論理関数設計方法の主要な点は多入力多出力の論理関数間の計算が母関数に基づく式により明解に計算できる点である。

- 5 母関数の応用は非常に多岐に亘る為、ここでは母関数による方法の代表的な 1 実施例として 2 入力 3 出力の論理関数 A と 3 入力 2 出力の論理関数のそれぞれの成分関数が与えられた時に A を B に代入して合成した 2 入力 2 出力の論理関数 C の成分論理関数をプログラムで求める例で示す。

- 10 この実施例では論理関数は入力を配列のアドレスに対応させ、出力は最小項に対応し、値 0, 1 で表す場合で示す。

式に基づく計算のプログラム化は容易なので処理の流れだけを以下に示す。

このプログラムに用いた配列は

- a [3][4]: 2 入力 3 出力論理関数 A ; 3 出力毎の成分論理関数の配列
 b [2][8]: 3 入力 2 出力論理関数 B ; 2 出力毎の成分論理関数の配列
 15 c [2][4]: 2 入力 2 出力論理関数 C ; 2 出力毎の成分論理関数の配列
 m a [8][4]: A の母関数配列 ;
 m b [4][8]: B の母関数配列 ;
 m c [4][4]: C の母関数配列

a [3][4]、b [2][8] には値 0, 1 が既に入っている。

- 20 処理の流れは

(ステップ 1) 成分論理関数 a [3][4]、b [2][8] から母関数 m a [8][4], m b [4][8] を計算

請求項 6 の母関数の定義式 (16) に従い計算する。

- (ステップ 2) 母関数 m a [8][4], m b [4][8] から母関数 m c [4][4]
 25 を計算

請求項 6 の (17) 式に従い計算する。

(ステップ 3) 母関数 m c [4][4] から成分論理関数 c [2][4] を計算

請求項 6 の (20) 式に従い計算する。

以上のように大きく 3 ステップの簡潔なプログラムにより計算ができる。

このように請求項6の発明の論理関数設計方法によれば全て明解に定義された式に基づくプログラム化が可能で論理関数設計において簡潔で効率的な設計が可能となる。

5 [実施例6]

次に請求項7の発明の実施例を以下に示す。

請求項7の発明の論理関数設計方法の主要な点は変数依存度を明解な式に基づき効率的に計算できる点である。

ここでは与えられた100個の4入力論理関数の中から指定された変数に依存しない論理関数を選別する問題につき、請求項7の(34)式に基づく決定プログラムの処理の流れだけを示す。用いる定数と変数は以下で論理関数配列は最小項に対応しアドレスに1が入りその他0の表現である。

f [100][16]: 100個の4ビット論理関数; 与えられた既知定数

f n [100][16]: 上記100個の否定論理関数: 0, 1が上と逆転

15 e [4]: 指定変数依存; e [u] = θ^u なる値が入った定数配列

d e l t a [16][16]: 変数依存度関数; (34)式の $\Delta(X,Y)$ に対応するものの

d [100]: 判定結果配列;

s [4][16]: 4標準成分論理関数配列; $x_0 \sim x_3$ の4つの論理関数に対応した

20 定数配列

s n [4][16]: 4標準成分否定論理関数配列: $\overline{x_0} \sim \overline{x_3}$ の論理関数に対応した定数配列

i, j, k: 一般の変数

(ステップ1)与えられたe [4]から変数依存度関数d e l t a [16][16]

25 を計算

請求項7に示された $\Delta(X,Y)$ の定義に基づきd e l t a [16][16]を計算
k = 0とする。

(ステップ2)(34)式の左辺の計算

$f[k][i] * f_n[k][j] * \text{delta}[i][j]$ を全ての i, j につき論理和計算を行う

計算結果が 0 なら $d[k] = 1$, その他は $d[k] = 0$

$k = k + 1$ とし $k = 100$ ならステップ 3、その他ステップ 2

5 (ステップ 3) 論理関数選別

$d[k] = 1$ なる k の論理関数を選別する。

以上が請求項 7 の発明の実施例である。請求項 7 の発明の論理関数設計方法によれば明解に定義された式により予め指定された変数依存度関数を計算することや、母関数による一括解析などが可能となり、従来の個別の変数の独立性を逐次調べる方法や、成分単位に調べる方法に比べ格段に効率的な処理になる。

[実施例 7]

請求項 8 の発明の主要な点は請求項 5 乃至請求項 7 における計算および判別を包括的に行うことでより多様な論理関数設計に用いることができる方法である。

- 15 図 6 は請求項 8 の発明のプログラム構成図で、以下のステップ 1 からステップ 7 までの 7 つの処理のステップと、ステップ間の処理の流れ制御するステップ 8 の大きくは 8 つのステップを有する。

[ステップ S 2 1]: 特徴論理関数変換ステップ。

- 20 入力された集合間の関係を関係式 (8) ~ (14) を用いて特徴論理関数の式に、変換する。

[ステップ S 2 2]: 特徴論理関数関係判別ステップ。

変換された特徴論理関数の式が成立するかどうかを判別する。

[ステップ S 2 3]: 写像成分関数・母関数入力ステップ。

各写像の成分関数または／および母関数を入力する。

- 25 [ステップ S 2 4] 母関数計算ステップ。

(16) 式に基づき母関数を計算する。

[ステップ S 2 5]: 成分関数計算ステップ。

(20) 式に基づき成分関数を計算する。

[ステップ S 2 6]: 成分関数・母関数関係計算ステップ。

(18) ~ (19) 式を計算する。

[ステップ S 27]: 変数依存度解析ステップ。

(30) ~ (35) 式に基づき変数依存度を解析する。

[ステップ S 28]: 結果を出力する。

5 [ステップ S 29]: データ制御ステップ。

ステップ S 21 ~ ステップ S 28 の間のデータの受け渡しを目的の処理に応じ制御する。

この図 6 の構成図に基づき設計する場合の具体例を以下に説明する。

10 2^m 人のユーザの K 項目についての好き嫌いを 2 値で表した集合データ 1 があり、この K 項目の好き嫌いのパターン 2^K 種の内、L 種のパター (集合データ 2) に該当する人をピックアップする論理回路、つまり m ビットのユーザ番号を入力した時に、該当者では 1、非該当者は 0 を出力する回路の論理関数を図 6 の構成図に従い設計する場合で説明する。

15 全てステップ 8 の制御の下に処理は進み、先ず 2^m 人の集合データ 1 からステップ 1 により m 個の特徴論理関数に変換し、これを写像 A の m 個の成分関数としてステップ 4 により母関数 A を計算する。同様に L 要素の集合データ 2 からステップ 1 により 1 個のパターン選別を表す特徴論理関数に変換し、これを写像 B の 1 個の成分関数としてステップ 4 により母関数 B を計算する。次に母関数 A と母関数 B をステップ 6 により合成写像 B A の母関数 B A を計算し、さらに同じステップ 6 により母関数 B A から 1 つの成分関数を計算する。

このようにして求める回路の論理関数が得られる。

この具体例はステップ S 21、ステップ S 24、ステップ S 26、ステップ S 28 だけの処理で設計できる例を示したが、その他のステップの用いる例は多岐に亘り存在する。

25 以上のように請求項 8 の発明の論理関数設計方法を用いれば多岐に亘る応用において、請求項 5 乃至請求項 7 の発明における計算および判別を包括的に行うことでより多様な論理関数設計行うことができる。

次に本発明の符号化演算方式の効果を顕著に示す具体例を説明する。

[具体例 1]

4 要素からなる集合 $\Omega^1_{in} = (s, t, u, v,)$ を入力、5 要素からなる集合 $\Omega^1_{out} = (a, c, f, g, h)$ を出力とする 2 項演算 $F^1: \Omega^1_{in} \times \Omega^1_{in} \rightarrow \Omega^1_{out}$ と、8 要素からなる集合 $\Omega^2 = (a, b, c, d, e, f, g, h)$ を入出力とする 2 項演算 $F^2: \Omega^2 \times \Omega^2 \rightarrow \Omega^2$ が定義された旧演算系を考える。本発明の効果を理解し易いように、擬似的に F^1 を乗算、 F^2 を加算と見なした時の積和演算（擬似的な積和演算）を実行する元演算系の計算システムにつき元演算系と新演算系を後に比較する。

10 F^1 は擬似乗算で $*$ で表し、 F^2 は擬似加算で $+$ で表し、複数の擬似加算を通常と同じように Σ で表すことにすると、擬似積和演算は

$$(58) \quad Z = \sum_{k=1}^K X_k * Y_k \text{ なる計算をすることになる。ここに } X_k, Y_k \in \Omega^1_{in}, Z \in \Omega^2$$

最も一般的な 2 値で元演算系を論理設計を行うことにすると、 $\Omega^1_{in} = B^2$ として、 $s \mapsto (0,0), t \mapsto (0,1), u \mapsto (1,0), v \mapsto (1,1)$ に対応させ、また $\Omega^1_{out} = \Omega^2 = B^3$ として $a \mapsto (0,0,0), b \mapsto (0,0,1), c \mapsto (0,1,0), d \mapsto (0,1,1), e \mapsto (1,0,0), f \mapsto (1,0,1), g \mapsto (1,1,0), h \mapsto (1,1,1)$ に対応させると、 F^1 は 4 ビット入力 3 ビット出力の 2 項演算器で、 F^2 は 6 ビット入力 3 ビット出力の 2 項演算器となる。この時の F^1 の真理値表を図 7 に、 F^2 の真理値表を図 8 に示す。さらにこの図 7 と図 8 の真理値表から論理合成ソフトにより合成した回路の回路図を示したものが図 9、図 10 である。図 9 は F^1 の回路図、図 10 は F^2 の回路図である。

20 この具体例 1 では、上述の発明の第 1 の側面に従う手法に基づき、元データ集合を全て旧表現データの集合 B^3 に拡張し、元演算系の F^1, F^2 に対応した旧演算 F^1_o, F^2_o を $F^1_o, F^2_o: B^3 \times B^3 \rightarrow B^3$ なる 2 項演算と考え、また新表現データの集合を同じ B^3 とする。この時符号化は同型写像による非冗長符号になる。つまり $\Phi: B^3 \rightarrow B^3$ 、 $\Psi = \Phi^{-1}: B^3 \rightarrow B^3$ となる。具体例 1 では図 11 のフローに従い新演算系を決定を行った。

図 11 の処理は次のようなものである。

[ステップ S 2 1]: $k = 1$ 、 $p = 1$ に初期設定する。 k は新演算器の入力数、 p は新演算器のビット数である。

[ステップ S 2 2]: p 番目の符号 Φ を生成する (置換 Φ を決める)。

[ステップ S 2 3]: 疑似加算器の真理値表を Φ で置換し新演算器を生成する。

[ステップ S 2 4]: k 変数依存以下の 2 項演算器を生成する。

[ステップ S 2 5]: 新疑似乗算器候補すべてにたいして Φ の逆置換を施し、元疑

5 似乗算との対応点を比較する。

[ステップ S 2 6]: 一致する候補がある化判別する。あればステップ S 2 7 へ進み、なければステップ S 2 8 へ進む。

[ステップ S 2 7]: 新演算器は満足するものかを判別する。満足するものであれば終了し、そうでなければステップ S 2 8 へ進む。

10 [ステップ S 2 8]: $p = P$ を判別し、 $p = P$ でなければ、ステップ S 2 9 へ進む。

[ステップ S 2 9]: $p = p + 1$ としてステップ S 2 2 に戻り処理を繰り返す。

[ステップ S 3 0]: $p = 1$ とする。

[ステップ S 3 1]: $k = K$ を判別し、 $k = K$ でなければステップ S 3 2 へ進む。
 $k = K$ であれば処理を終了する。

15 [ステップ S 3 2]: $k = k + 1$ としてステップ S 2 2 へ戻り、処理を繰り返す。

図 1 1 の処理の結果得られた元演算系の F^1, F^2 に対応した一組 (同等のものは複数個存在する) の新演算器を F^1_N, F^2_N とすると、この F^1_N, F^2_N は

$F^1_N, F^2_N: B^3 \times B^3 \rightarrow B^3$ なる演算 (写像) となる。これらの真理値表を示したも

20 のが図 1 2 と図 1 3 である。また図 1 4 と図 1 5 はこの符号化における符号器と復号器の真理値表である。

これらの図 1 2、図 1 3、図 1 4、図 1 5 の真理値表から論理合成ソフトにより合成した回路の回路図を示したものが図 1 6、図 1 7、図 1 8、図 1 9 である。

ここで図 1 1 の新演算系決定フローを説明する。この具体例 1 では符号化は非
25 冗長符号化であり、符号器 Φ は同型写像となり、 B^3 の 8 種の要素 (データ) を置換することに対応する。この置換は 8 ! 種存在する。 $P = 8 !$ とし、 P 種の置換を番号付けする。元疑似加算と旧演算系の疑似加算はどちらも B^3 上の 2 項演算なので、同じものとして扱い、真理値表に対して入力と出力の 3 ビットの値に対し置換施し入力を順に並べ替えたものが図 1 3 で、これは符号化 (置換) により生

成された擬似加算の新演算器の真理値表になる。また元擬似乗算は旧演算系への拡張を直接的に行わずに、先ず新演算系の擬似乗算器の候補を出し、各符号化に対応して逆の置換 Φ^{-1} を施し、候補に対応する旧演算系の擬似乗算を生成し、対応する部分が元擬似乗算の真理値表と一致するかどうかを判定する。つまり一致した新演算系の擬似乗算の候補が、元擬似乗算を拡張した旧演算系の擬似乗算を符号化したものに相当する。

以上の具体例 1 において擬似乗算に関しては図 9 と図 16、擬似加算に関しては図 10 と図 17 を比較すれば解るように新演算系の演算器の回路が格段に簡単になっていることが解る。さらに具体的な効果を観る意味で、NOT（インバータ）の素子数を \square 、遅延時間を Δ とし、演算器単体の素子数とクリティカルパスの遅延時間を F^1, F^2, F^1_N, F^2_N の順にそれぞれ $\square^1, \square^2, \square^1_N, \square^2_N$ 、と $\Delta^1, \Delta^2, \Delta^1_N, \Delta^2_N$ とし、さらに Φ, Ψ の素子数とクリティカルパスの遅延時間を $\square_\Phi, \square_\Psi$ と Δ_Φ, Δ_Ψ とすると、図 20 の具体例 1 の素子数遅延表のようになる。ここに各ゲートの素子数と遅延時間は一般的なものである 2 入力 and ゲートと 2 入力 or ゲートの素子数、遅延時間は $2\square, 2\Delta$ とし、2 入力 xor ゲートの素子数、遅延時間は $4\square, 3\Delta$ とした。

ここでさらに (58) の擬似積和演算を専用のハードウェアで実現した場合で示す。

図 21 の木構造積和演算器ブロック図に示された木構造に配列された専用積和演算回路を、* が付された乗算器のところに擬似乗算器を、+ が付された加算器のところに擬似加算器を配列した元演算系と新演算系の擬似積和演算器を構成し評価してみる。

図 21 は (58) の $K=8=2^3$ に相当するブロック図であるが、 $K=2^L$ (L は自然数) なる場合には記号 * の乗算器は K 個並列に並び、これらの乗算器の出力に続く記号 + の加算器は格段毎に 2^{L-1} 個、 2^{L-2} 個、 \dots 、2、1 個が接続された木構造を成す。従って記号 * の乗算器の素子数を \square_* 、遅延時間を Δ_* 、記号 + の加算器の素子数を \square_+ 、遅延時間を Δ_+ と表すと

$$(59) \text{ 積和演算器素子数} = 2^L \cdot \square_* + (2^L - 1) \cdot \square_+$$

$$(60) \text{ 積和演算器遅延時間} = \Delta_* + L \cdot \Delta_+$$

となる。 $K=8=2^3$ の場合には素子数は元演算系が2325□、新演算系が100□、遅延時間は元演算系が100△、新演算系が21△であり、擬似積和演算器だけで比較すると新演算系は元演算系の素子数で約1/23、遅延時間は約1/5になっている。

- 5 実際には新演算系に対しては符号器と復号器が追加して評価すべきである。符号器、復号器の数などは各種考えられるが、ここでは図22に示す回路で比較する。

- 図22は $2K=16$ 個の入力データ $X_1, Y_1, \dots, X_8, Y_8$ 毎に1つの符号器 Φ を配し、それぞれを $X'_1, Y'_1, \dots, X'_8, Y'_8$ なる新表現データに変換し、図21の構造を有した新演算系擬似積和演算器で新表現データとしての積和演算結果 Z' 計算し、復号器 Ψ で元演算系の擬似積和演算結果 Z を出力する本発明の符号化演算方式の1つの実施回路である。この回路Aは、内部をブラックボックスとしてみると前記の図21の構造を有した $X_1, Y_1, \dots, X_8, Y_8$ から Z を算出する元演算系の擬似積和演算器と機能としては全く同じであるが、素子数が
- 10 467□、遅延時間が36△に大幅に削減される。

- 以上のように本発明の符号化演算方式によれば閉じた演算系に限らず、一般的な演算系（元演算系）に対し、大幅に素子数や遅延時間を削減した演算装置を実現することができる。またこの具体例では基数 $r=2$ の場合で示したが、多値論理の場合も基本的に同じ流れで符号化演算方式を実現できることは言うまでも無い。
- 20

尚、図11のフローは本発明に基づく論理設計のあくまでも1つのアルゴリズムであり、幾つもの変形したアルゴリズムが考えられる。

- また図21に示した具体例1の木構造積和演算ブロック図の構成は、擬似積和演算に限らず例えば浮動小数点演算のような演算器に対しても同様な構成が可能であり、図22は符号器、復号器および演算器からなる演算処理装置全体を示している。
- 25

またCPUまたはデータ制御装置とを具備し、1つまたは複数の新演算器による演算結果を復号器により旧表現データに変換すること無く、新表現データのままで一旦メモリに格納し、他の計算プロセスにおいて計算された新表現データを

前記CPUまたはデータ制御装置により前記結果の新表現データとの計算を必要な計算手順で計算可能にする演算処理装置も実現可能である。

〔具体例 2〕

- 5 次に母関数を利用した論理関数設計方式を分かり易い例として B^2 で定義された 2 つの単項演算 G^1 と G^2 だけからなる元演算系の場合の具体例で示す。図 2 3 は G^1 と G^2 の真理値表である。この単項演算 G^1 と G^2 は $G^1, G^2: B^2 \rightarrow B^2$ なる写像であり、その成分関数を $g^1_0(X)$ 、 $g^1_1(X)$ および $g^2_0(X)$ 、 $g^2_1(X)$ とすると、真理値表から、 $g^1_0(X) = x_1$ 、 $g^1_1(X) = x_1 \cdot x_0$ および $g^2_0(X) = x_1 \cdot \overline{x_0}$ 、 $g^2_1(X) = x_1 \cdot x_0$ であり、(素子数、遅延時間) は G^1 は (2 □、2 □) と G^2 は (5 □、3 △) となる。
- 10 また G^1 と G^2 の母関数 $\tilde{G}^1(Y, X)$ と $\tilde{G}^2(Y, X)$ は母関数の定義式 (1 6) より

$$\begin{aligned}
 \tilde{G}^1(Y, X) &= (y_0 \cdot g^1_0(x) \cup \overline{y_0} \cdot \overline{g^1_0(X)}) \cdot (y_1 \cdot g^1_1(x) \cup \overline{y_1} \cdot \overline{g^1_1(X)}) \\
 (6 \ 1) \quad &= (y_0 \cdot x_1 \cup \overline{y_0} \cdot \overline{x_1}) \cdot (y_1 \cdot x_1 \cdot x_0 \cup \overline{y_1} \cdot \overline{x_1 \cdot x_0}) \\
 &= Y^0 \cdot X^0 \cup Y^0 \cdot X^1 \cup Y^1 \cdot X^2 \cup Y^3 \cdot X^3
 \end{aligned}$$

同様に

$$(6 \ 2) \quad \tilde{G}^2(Y, X) = Y^0 \cdot X^0 \cup Y^0 \cdot X^1 \cup Y^1 \cdot X^2 \cup Y^2 \cdot X^3$$

- 15 となる。これらの母関数をブール値行列として図示したものが図 2 4 である。

この元演算系は B^2 上の同型写像の範囲の符号化ではこれ以上簡単にならない。そこでこの具体例 2 では非符号無しの 3 ビットの冗長符号化まで考え新演算系を設計した。非符号無しの時には符号領域の特徴論理関数 $c(X')$ ($X' \in B^3$) は $c(X') = 1$ であるから、先ず符号器、復号器の関係式 (1 - b) は以下のような同値な変形が成り立つ。

20

$$\begin{aligned}
 (6 \ 3) \quad &\tilde{\Psi}(X, X') \cdot \tilde{\Phi}(X', X) = \tilde{\Phi}(X', X) \\
 &\Leftrightarrow \tilde{\Psi}(X, X') \geq \tilde{\Phi}(X', X)
 \end{aligned}$$

この (6 3) と (2 - b) より、この非符号無しの符号設計問題は 4 種の各 X に対応する $\tilde{\Psi}(X, X')$ を B^3 上の 8 種類の最小項を入れる箱として考え、これらの 4 つの箱に重複が無く且つ空の箱が生じないように 8 種の最小項を箱詰めする問

題と等価になる。そしてこの箱詰めの方法が1つ決まれば $\tilde{\Psi}(X, X')$ が1つ決まることに対応する。

また (3-b-1) の相当する式は、以下のように同値な変形が成り立つ。

$$\begin{aligned}
 & \overline{\tilde{G}^q_o(Y, X)} \cdot \tilde{G}^q_N(Y', X') \cdot \tilde{\Psi}(Y, Y') \cdot \tilde{\Psi}(X, X') = 0 \quad (Y' \in B^3, q=1,2) \\
 (64) \quad & \Leftrightarrow \overline{\tilde{G}^q_o(\Psi(X'), \Psi(Y'))} \cdot \tilde{G}^q_N(Y', X') = 0 \\
 & \Leftrightarrow \tilde{G}^q_N(Y', X') \leq \tilde{G}^q_o(\Psi(Y'), \Psi(X')) \\
 & \Leftrightarrow \tilde{G}^q_N(Y', X') \leq \bigcup_Y \bigcup_X \tilde{G}^q_o(Y, X) \cdot \tilde{\Psi}(Y, Y') \cdot \tilde{\Psi}(X, X')
 \end{aligned}$$

- 5 また (3-b-2) は $c(X')=1$ では常に成り立つ。そこでこの具体例2では図25にフローに従い新演算系の2つの単項演算を簡単なものから順に生成し、(64)の最後の式を満足するかを判定しながら新演算系の決定を行った。

図25の処理は以下のようなものである。

[ステップS41]: 簡単な新演算器候補を生成する。

- 10 [ステップS42]: 新演算器の母関数を構成する。

[ステップS43]: 箱詰めにより符号を生成する。

[ステップS44]: (64)式を満たすかどうかを判別し、満たせば処理を終了し、満たさなければステップS45へ進む。

- 15 [ステップS45]: すべての符号を生成したかを判別し、生成した場合にはステップS46へ進み、そうでない場合にはステップS43へ戻り、処理を繰り返す。

[ステップS46]: 次の新演算器候補を生成してステップS42へ戻り、処理を繰り返す。

図25の処理の結果は、

$$(65) \quad \tilde{\Psi}(X, X') = X^0 \cdot X'^6 \cup X^1 \cdot X'^2 \cup X^2 \cdot (X'^0 \cup X'^4) \cup X^3 \cdot (X'^1 \cup X'^3 \cup X'^5 \cup X'^7)$$

- 20 つまり、箱 $\tilde{\Psi}(0, X')$ には1つの最小項 X'^6 を、箱 $\tilde{\Psi}(1, X')$ には1つの最小項 X'^2 を、 $\tilde{\Psi}(2, X')$ には2つの最小項 X'^0 と X'^4 を、箱 $\tilde{\Psi}(3, X')$ には4つの最小項 X'^1 、 X'^3 、 X'^5 、 X'^7 を箱詰めした場合に相当する符号化に対して、(64)の最下段を満足する新演算系の単項演算 G^1_N と G^2_N の成分関数が $g^1_{N0}(X') = x'_0$ 、 $g^1_{N1}(X') = \overline{x'_0}$ 、 $g^1_{N2}(X') = x'_1$

$g^2_{N0}(X')=0$ 、 $g^2_{N1}(X')=\overline{x'_0}$ 、 $g^2_{N2}(X')=\overline{x'_1}$ 、となるものが求まった。(素子数、遅延時間)は G^1_N は(□、△)、 G^2_N は(2□、△)となり、元演算系の G^1 に比較し大幅に簡単になっている。尚、この場合(63)式を満足する母関数 $\tilde{\Phi}(X',X)$ は全て符号器の母関数となる。例えば

$$5 \quad (66) \quad \tilde{\Phi}(X',X)=X'^6 \cdot X^0 \cup X'^2 \cdot X^1 \cup X'^0 \cdot X^2 \cup X'^1 \cdot X^3$$

はこの具体例の符号器の母関数の例である。

ここで前記 Ψ および G^1_N が(64)式を満足することを確認すると、(64)の左辺は

$$\begin{aligned} \tilde{G}^1_N(Y',X') &= (y'_0 \cdot x'_0 \cup \overline{y'_0} \cdot \overline{x'_0}) \cdot (y'_1 \cdot \overline{x'_0} \cup \overline{y'_1} \cdot x'_0) \cdot (y'_2 \cdot x'_1 \cup \overline{y'_2} \cdot \overline{x'_1}) \\ &= Y'^2 \cdot X'^0 \cup Y'^1 \cdot X'^1 \cup Y'^6 \cdot X'^2 \cup Y'^5 \cdot X'^3 \cup Y'^2 \cdot X'^4 \cup Y'^1 \cdot X'^5 \cup Y'^6 \cdot X'^6 \cup Y'^5 \cdot X'^7 \end{aligned}$$

10 であり、一方(64)の右辺は(61)と(65)より

$$\begin{aligned} &\bigcup_{Y'} \bigcup_{Y'} \tilde{G}^1(Y,X) \cdot \tilde{\Psi}(Y,Y') \cdot \tilde{\Psi}(X,X') \\ &= \tilde{\Psi}(0,Y') \cdot \tilde{\Psi}(0,X') \cup \tilde{\Psi}(0,Y') \cdot \tilde{\Psi}(1,X') \cup \tilde{\Psi}(1,Y') \cdot \tilde{\Psi}(2,X') \cup \tilde{\Psi}(3,Y') \cdot \tilde{\Psi}(3,X') \\ &= Y'^6 \cdot X'^6 \cup Y'^6 \cdot X'^2 \cup Y'^2 \cdot (X'^0 \cup X'^4) \cup (Y'^1 \cup Y'^3 \cup Y'^5 \cup Y'^7) \cdot (X'^2 \cup X'^3 \cup X'^5 \cup X'^7) \end{aligned}$$

となることから(64)式は明らかに成り立つ。

ここで具体例2における演算器のデータの入出力の関係をトポロジーとして表してみる。図26は元演算器 G^1 と G^2 の入出力トポロジーを表したもので、 G^1 につき説明すると、図23より G^1 は $0=(0,0)$ の入力は出力 $0=(0,0)$ を出し、 $3=(1,1)$ の入力は出力 $3=(1,1)$ を出し、 $0=(0,0)$ と $3=(1,1)$ は入出力の関係はトポロジーとして観た時に自己ループしている。また入力 $1=(0,1)$ は $0=(0,0)$ を出力し、入力 $2=(1,0)$ は $1=(0,1)$ を出力していることを図26は表している。この入出力関係のトポロジーと真理値表は完全に対応が付く。 G^2 についても同様である。また図27は新演算器 G^1_N と G^2_N につき同様に入出力トポロジーを表したものである。図26と図27とはトポロジーとして全く異なっているが、符号化により、 B^3 上で $0=(0,0,0)$ と $4=(1,0,0)$ は同じ符号[2]に属し、また $1=(0,0,1)$ 、 $3=(0,1,1)$ 、 $5=(1,0,1)$ 、 $7=(1,1,1)$ は同じ符号[3]に属することから、これらを図中の楕円で囲んだように束ね一つのグループと見なすとグループ間の入出力関係は図28のようになる。そして図26と図28はトポロジーとして同型であることが分かる。このように冗長

符号化の本質は前記のように同じ符号を束ねた時の新演算器のトポロジーが元演算系の演算器のトポロジーと同型になることであると言えることができる。そのことから、トポロジーを同型にするように新演算器を設計することが可能である。このトポロジーを利用した新演算系の決定のフローの具体例を示したものが図 29（トポロジー一致型新演算系決定フロー）で、これは $r = 2$ に限定されず、多値論理式として設計する場合にも有効な方法である。

図 29 の処理は以下ようになる。

[ステップ S 5 1]: 元演算器の入出力関係のトポロジーを作成する。

[ステップ S 5 2]: 簡単な新演算器後方を生成する。

10 [ステップ S 5 3]: 新演算器の入出力関係のトポロジーを作成する。

[ステップ S 5 4]: 箱詰めにより符号を生成する。

[ステップ S 5 5]: 符号に降り束ねた新演算器の入出力関係のトポロジーを生成する。

15 [ステップ S 5 6]: 元演算系と新演算系がトポロジーとして同型かどうかを判別する。同型であれば処理を終了する。同型でなければステップ S 5 7 へ進む。

[ステップ S 5 7]: すべての符号を生成したかを判別する。すべての符号を生成していないと判別したときにはステップ S 5 4 へ戻り処理を繰り返す。すべての符号を生成したと判別した場合にはステップ S 5 8 へ進む。

20 [ステップ S 5 8]: 次の新演算器候補を生成してステップ S 5 3 へ戻り処理を繰り返す。

25 以上のように具体例 2 では母関数に対して成り立つ $(1 - b)$ 、 $(2 - b)$ 、 $(3 - b - 1)$ 、 $(3 - b - 2)$ と同値な式または関係を用い元演算系の情報から非冗長符号化による新演算系を論理設計を行うことができた。全く同様に $(1 - b)$ 乃至 $(3 - b - 2)$ またはこれらと同値な式に基づき 2 項演算や T 項演算を含む元演算系から新演算系を論理設計することができる。

以上で実施例およびその具体例の説明を終了する。

なお、この発明は上述の実施例に限定されるものではなくその趣旨を逸脱しない範囲で種々変更が可能である。

請求の範囲

1. 符号器と、上記符号器の出力を演算処理する演算処理手段と、上記演算処理手段の出力を復号する復号器とを有し、第1の表現データ上で定義された元の演算系の処理を、第2の表現データ上で定義された上記演算処理手段の新たな演算系の処理で代替する演算処理装置において、

単項演算型の Q 種の演算 $G^q: \Omega_{G^q}in \rightarrow \Omega_{G^q}out$ ($q=1,2,\dots,Q$) および／または 2

項演算型の P 種の演算 $F^p: \Omega_{F^p}in \times \Omega_{F^p}in \rightarrow \Omega_{F^p}out$ ($p=1,2,\dots,P$) および／また

は T 項演算型の S 種の演算 $H^s: \Omega_{H^s}in \times \Omega_{H^s}in \times \dots \times \Omega_{H^s}in \rightarrow \Omega_{H^s}out$ ($s=1,2,\dots,S$)

が 1 つまたは複数個 ($Q+P+S \geq 1$) 定義された元の演算系の入力空間の有限

10 集合 $\Omega_{G^q}in$ 、 $\Omega_{F^p}in$ 、 $\Omega_{H^s}in$ と出力空間の有限集合 $\Omega_{G^q}out$ 、 $\Omega_{F^p}out$ 、 $\Omega_{H^s}out$ の

要素数 $|\Omega_{G^q}in|$ 、 $|\Omega_{F^p}in|$ 、 $|\Omega_{H^s}in|$ 、 $|\Omega_{G^q}out|$ 、 $|\Omega_{F^p}out|$ 、 $|\Omega_{H^s}out|$ とに対して、

$\max\{|\Omega_{G^q}in|, |\Omega_{F^p}in|, |\Omega_{H^s}in|, |\Omega_{G^q}out|, |\Omega_{F^p}out|, |\Omega_{H^s}out|\} \leq r^n$ を満たす基数 r 、

語長 n の数表現の集合 B_r^n (r 値の集合 B_r の n 個の直積) を、元の演算系の第 1

の表現データの集合とし、前記の元の演算系の各演算の入力空間のデータ集合

15 Ωin ($\Omega_{G^q}in$ 、 $\Omega_{F^p}in$ 、 $\Omega_{H^s}in$ の何れか) の要素数 $|\Omega in|$ に対して、 $|\Omega in| = r^n$ なる時

以外は、 $r^n - |\Omega|$ 個分の定まっていない要素に対する対応関係を元の演算系の各

演算に付加し、元の演算系の Q 種の単項演算型の演算 G^q は

$G^q_o: B_r^n \rightarrow B_r^n$ ($q=1,2,\dots,Q$) なる単項演算に拡張し、P 種の二項演算型の演算 F^p

は $F^p_o: B_r^n \times B_r^n \rightarrow B_r^n$ ($p=1,2,\dots,P$) なる 2 項演算に拡張し、S 種の T 項演算型

20 の演算 H^s は $H^s_o: B_r^n \times B_r^n \times \dots \times B_r^n \rightarrow B_r^n$ (直積は T 個、 $s=1,2,\dots,S$) なる T

項演算に拡張し、さらに第 2 の表現データは集合 B_r^m ($m \geq n$) 上のデータとした

ときに、

上記符号器は $\Phi: B_r^n \rightarrow B_r^m$ なる単射の写像として動作し、

上記復号器は $\Psi: B_r^m \rightarrow B_r^n$ なる全射の写像として動作し、

25 上記演算処理手段は、さらに G^q_o に対応する新たな演算系の単項演算である

$G^q_N: B_r^m \rightarrow B_r^m$ として動作し、 F^p_o に対応する新たな演算系の二項演算である

$F^p_N: B_r^m \times B_r^m \rightarrow B_r^m$ として動作し、さらに、 H^s_o に対応する新たな演算系の T 項演算である $H^s_N: B_r^m \times B_r^m \times \cdots \times B_r^m \rightarrow B_r^m$ として動作し、

もって、元の演算系の全ての演算と新たな演算系の全ての演算および符号器、復号器を全て多入力多出力の r 値論理の写像に対応させ、

- 5 さらに、 B_r^n 上の任意 X に対応した符号 $[X]$ ($[X] \subset B_r^m$) に対し
- (1) $\Phi(X) \in [X] \subset B_r^m$ (for $\forall X \in B_r^n$)
 - (2) $\Psi([X]) = X$ (for $\forall X \in B_r^n$)
 - (3) $Y = G^q_o(X) \Leftrightarrow [Y] \supset G^q_N([X])$ (for $\forall X, Y \in B_r^n, \forall q$)
 - (4) $Z = F^p_o(X, Y) \Leftrightarrow [Z] \supset F^p_N([X], [Y])$ (for $\forall X, Y, Z \in B_r^n, \forall p$)
 - 10 (5) $Y = H^s_o(X_1, \dots, X_T) \Leftrightarrow [Y] \supset H^s_N([X_1], \dots, [X_T])$ (for $\forall X_1, \dots, X_T, Y \in B_r^n, \forall s$)

の条件を満たすようにしたことを特徴とする演算処理装置。

2. 符号器と、上記符号器の出力を演算処理する演算処理手段と、上記演算処理手段の出力を復号する復号器とを有し、第 1 の表現データ上で定義された元の演算系の処理を、第 2 の表現データ上で定義された上記演算処理手段の新たな演算系の処理で代替する演算処理装置において、
- 15

単項演算型の Q 種の演算 $G^q: \Omega_{G^q}in \rightarrow \Omega_{G^q}out$ ($q = 1, 2, \dots, Q$) および／または 2

項演算型の P 種の演算 $F^p: \Omega_{F^p}in \times \Omega_{F^p}in \rightarrow \Omega_{F^p}out$ ($p = 1, 2, \dots, P$) および／また

- 20 は T 項演算型の S 種の演算 $H^s: \Omega_{H^s}in \times \Omega_{H^s}in \times \cdots \times \Omega_{H^s}in \rightarrow \Omega_{H^s}out$ ($s = 1, 2, \dots, S$) が 1 つまたは複数個 ($Q + P + S \geq 1$) 定義された元の演算系の入力空間の有限集合 $\Omega_{G^q}in$ 、 $\Omega_{F^p}in$ 、 $\Omega_{H^s}in$ と出力空間の有限集合 $\Omega_{G^q}out$ 、 $\Omega_{F^p}out$ 、 $\Omega_{H^s}out$ の要素数 $|\Omega_{G^q}in|$ 、 $|\Omega_{F^p}in|$ 、 $|\Omega_{H^s}in|$ 、 $|\Omega_{G^q}out|$ 、 $|\Omega_{F^p}out|$ 、 $|\Omega_{H^s}out|$ とに対して、 $\max\{|\Omega_{G^q}in|, |\Omega_{F^p}in|, |\Omega_{H^s}in|, |\Omega_{G^q}out|, |\Omega_{F^p}out|, |\Omega_{H^s}out|\} \leq r^n$ を満たす基数 r、
- 25 語長 n の数表現の集合 B_r^n (r 値の集合 B_r の n 個の直積) を、元の演算系の第 1 の表現データの集合とし、前記の元の演算系の各演算の入力空間のデータ集合 Ωin ($\Omega_{G^q}in$ 、 $\Omega_{F^p}in$ 、 $\Omega_{H^s}in$ の何れか) の要素数 $|\Omega in|$ に対して、 $|\Omega in| = r^n$ なる時

以外は、 $r^n - |\Omega|$ 個分の定まっていない要素に対する対応関係を元の演算系の各演算に付加し、元の演算系の Q 種の単項演算型の演算 G^q は

$G^q_o: B_r^n \rightarrow B_r^n$ ($q=1,2,\dots,Q$) なる単項演算に拡張し、 P 種の二項演算型の演算 F^p

は $F^p_o: B_r^n \times B_r^n \rightarrow B_r^n$ ($p=1,2,\dots,P$) なる 2 項演算に拡張し、 S 種の T 項演算型

- 5 の演算 H^s は $H^s_o: B_r^n \times B_r^n \times \dots \times B_r^n \rightarrow B_r^n$ (直積は T 個、 $s=1,2,\dots,S$) なる T 項演算に拡張し、さらに第 2 の表現データは集合 B_r^m ($m \geq n$) 上のデータとしたときに、

上記符号器は $\Phi: B_r^n \rightarrow B_r^m$ なる単射の写像として動作し、

上記復号器は $\Psi: B_r^m \rightarrow B_r^n$ なる全射の写像として動作し、

- 10 上記演算処理手段は、さらに G^q_o に対応する新たな演算系の単項演算である

$G^q_N: B_r^m \rightarrow B_r^m$ として動作し、 F^p_o に対応する新たな演算系の二項演算である

$F^p_N: B_r^m \times B_r^m \rightarrow B_r^m$ として動作し、さらに、 H^s_o に対応する新たな演算系の T

項演算である $H^s_N: B_r^m \times B_r^m \times \dots \times B_r^m \rightarrow B_r^m$ として動作し、

もって、元の演算系の全ての演算と新たな演算系の全ての演算および符号器、

- 15 復号器を全て多入力多出力の r 値論理の写像に対応させ、

さらに、 B_r^n 上の任意 X に対応した符号 $[X]$ ($[X] \subset B_r^m$) に対し

$$(1c) \quad \chi'_{[X]}(\Phi(X)) = 1 \quad (\text{for } \forall X \in B_r^n)$$

$$(2c) \quad \chi_X(\Psi([X])) = 1 \quad (\text{for } \forall X \in B_r^n)$$

$$(3c) \quad \chi'_{[G^q_o(X)]}(G^q_N([X])) = 1 \quad (\text{for } \forall X \in B_r^n, \forall q)$$

- 20 (4c) $\chi'_{[F^p_o(X,Y)]}(F^p_N([X],[Y])) = 1 \quad (\text{for } \forall X, Y \in B_r^n, \forall p)$

$$(5c) \quad \chi'_{[H^s_o(X_1, \dots, X_T)]}(H^s_N([X_1], \dots, [X_T])) = 1 \quad (\text{for } \forall X_1, \dots, X_T \in B_r^n, \forall s)$$

の条件を満たすようにしたことを特徴とする演算処理装置。

3. 単項演算型の Q 種の演算 $G^q: \Omega_{G^q, in} \rightarrow \Omega_{G^q, out}$ ($q=1,2,\dots,Q$) および／また

は 2 項演算型の P 種の演算

$F^p: \Omega_{F^p}, in \times \Omega_{F^p}, in \rightarrow \Omega_{F^p}, out$ ($p=1,2,\dots,P$) および / または T 項演算型の S 種の

演算 $H^s: \Omega_{H^s}, in \times \Omega_{H^s}, in \times \dots \times \Omega_{H^s}, in \rightarrow \Omega_{H^s}, out$ ($s=1,2,\dots,S$) が 1 つまたは複数個

($Q+P+S \geq 1$) 定義された元の演算系に対し、入力空間の有限集合 Ω_{G^q}, in 、

5 Ω_{F^p}, in 、 Ω_{H^s}, in と出力空間の有限集合 Ω_{G^q}, out 、 Ω_{F^p}, out 、 Ω_{H^s}, out の要素数 $|\Omega_{G^q}, in|$ 、
 $|\Omega_{F^p}, in|$ 、 $|\Omega_{H^s}, in|$ 、 $|\Omega_{G^q}, out|$ 、 $|\Omega_{F^p}, out|$ 、 $|\Omega_{H^s}, out|$ の最大要素数に対し

$\max\{|\Omega_{G^q}, in|, |\Omega_{F^p}, in|, |\Omega_{H^s}, in|, |\Omega_{G^q}, out|, |\Omega_{F^p}, out|, |\Omega_{H^s}, out|\} \leq r^n$ を満たす基数 r 、

語長 n の数表現の集合 B_r^n (r 値の集合 B_r の n 個の直積) を元の演算系の第 1 の

表現データの集合とし、前記の元の演算系の各演算の入力空間のデータ集合

10 $\Omega in (\Omega_{G^q}, in, \Omega_{F^p}, in, \Omega_{H^s}, in \text{ の何れか})$ の要素数 $|\Omega in|$ に対して、 $|\Omega in| = r^n$ なる時

以外は、 $r^n - |\Omega|$ 個分の定まっていない要素に対する対応関係をそれぞれの元演

算毎に付加し、元の演算系の Q 種の単項演算型の演算 G^q は $G^q_o: B_r^n \rightarrow B_r^n$

($q=1,2,\dots,Q$) なる単項演算に拡張し、元の演算系の P 種の二項演算型の演算 F^p

は $F^p_o: B_r^n \times B_r^n \rightarrow B_r^n$ ($p=1,2,\dots,P$) なる 2 項演算に拡張し、元の演算系の S 種

15 の T 項演算型の元演算 H^s は $H^s_o: B_r^n \times B_r^n \times \dots \times B_r^n \rightarrow B_r^n$ (直積は T 個、
 $s=1,2,\dots,S$) なる T 項演算に拡張し、さらに集合 B_r^m ($m \geq n$) 上のデータを第

2 の表現データとし、符号器は $\Phi: B_r^n \rightarrow B_r^m$ なる単射の写像として扱い、復号

器は $\Psi: B_r^m \rightarrow B_r^n$ なる全射の写像として扱い、さらに G^q_o に対応する新演算系

の単項演算は $G^q_N: B_r^m \rightarrow B_r^m$ なる単項演算として扱い、 F^p_o に対応する新演算系

20 の二項演算は $F^p_N: B_r^m \times B_r^m \rightarrow B_r^m$ なる二項演算として扱い、 H^s_o に対応する

新演算系の T 項演算は $H^s_N: B_r^m \times B_r^m \times \dots \times B_r^m \rightarrow B_r^m$ なる T 項演算として扱うこ

とで、旧演算系の全ての演算と新演算系の全ての演算および符号器、復号器を全

て多入力多出力の r 値論理の写像に対応させ、

B_r^n 上の任意 X に対応して、

25 (1) $\Phi(X) \in [X] \subset B_r^m$ (for $\forall X \in B_r^n$)

(2) $\Psi([X]) = X$ (for $\forall X \in B_r^n$)

なる条件を満たす符号 $[X]$ ($[X] \subset B_n^m$) を生成するステップと、

新演算系の演算を生成するステップと、

上記演算のうち、

$$(3) Y = G^q_o(X) \Leftrightarrow [Y] \supset G^q_N([X]) \quad (\text{for } \forall X, Y \in B_r^n, \forall q)$$

$$5 \quad (4) Z = F^p_o(X, Y) \Leftrightarrow [Z] \supset F^p_N([X], [Y]) \quad (\text{for } \forall X, Y, Z \in B_r^n, \forall p)$$

$$(5) Y = H^s_o(X_1, \dots, X_T) \Leftrightarrow [Y] \supset H^s_N([X_1], \dots, [X_T]) \quad (\text{for } \forall X_1, \dots, X_T, Y \in B_r^n, \forall s)$$

なる条件を満たすものを選択するステップとをコンピュータに実行させることを特徴とする演算処理装置設計方法。

10

4. 単項演算型の Q 種の演算 $G^q: \Omega_{G^q}in \rightarrow \Omega_{G^q}out$ ($q = 1, 2, \dots, Q$) および／また

は 2 項演算型の P 種の演算 $F^p: \Omega_{F^p}in \times \Omega_{F^p}in \rightarrow \Omega_{F^p}out$ ($p = 1, 2, \dots, P$) および／

または T 項演算型の S 種の演算 $H^s: \Omega_{H^s}in \times \Omega_{H^s}in \times \dots \times \Omega_{H^s}in \rightarrow \Omega_{H^s}out$

($s = 1, 2, \dots, S$) が 1 つまたは複数個 ($Q + P + S \geq 1$) 定義された元の演算系の

15 入力空間の有限集合 $\Omega_{G^q}in$ 、 $\Omega_{F^p}in$ 、 $\Omega_{H^s}in$ と出力空間の有限集合 $\Omega_{G^q}out$ 、 $\Omega_{F^p}out$ 、 $\Omega_{H^s}out$ の要素数 $|\Omega_{G^q}in|$ 、 $|\Omega_{F^p}in|$ 、 $|\Omega_{H^s}in|$ 、 $|\Omega_{G^q}out|$ 、 $|\Omega_{F^p}out|$ 、 $|\Omega_{H^s}out|$ の最大要素数に対し

$$\max\{|\Omega_{G^q}in|, |\Omega_{F^p}in|, |\Omega_{H^s}in|, |\Omega_{G^q}out|, |\Omega_{F^p}out|, |\Omega_{H^s}out|\} \leq r^n$$

を満たす基数 r 、

20 語長 n の数表現の集合 B_r^n (r 値の集合 B_r の n 個の直積) を元の演算系の第 1 の表現データの集合とし、前記の元の演算系の各演算の入力空間のデータ集合

Ωin ($\Omega_{G^q}in$ 、 $\Omega_{F^p}in$ 、 $\Omega_{H^s}in$ の何れか) の要素数 $|\Omega in|$ に対して、 $|\Omega in| = r^n$ なる時

以外は、 $r^n - |\Omega|$ 個分の定まっていない要素に対する対応関係をそれぞれの元演

算毎に付加し、元の演算系の Q 種の単項演算型の演算 G^q は

$G^q_o: B_r^n \rightarrow B_r^n$ ($q = 1, 2, \dots, Q$) なる単項演算に拡張し、元の演算系の P 種の二項演

25 算型の演算 F^p は $F^p_o: B_r^n \times B_r^n \rightarrow B_r^n$ ($p = 1, 2, \dots, P$) なる 2 項演算に拡張し、元

の演算系の S 種の T 項演算型の元演算 H^s は $H^s_o: B_r^n \times B_r^n \times \dots \times B_r^n \rightarrow B_r^n$ (直積は T 個、 $s = 1, 2, \dots, S$) なる T 項演算に拡張し、さらに集合 B_r^m ($m \geq n$) 上のデ

ータを第2の表現データとし、符号器は $\Phi: B_r^n \rightarrow B_r^m$ なる単射の写像として扱い、復号器は $\Psi: B_r^m \rightarrow B_r^n$ なる全射の写像として扱い、さらに G^q_o に対応する新演算系の単項演算は $G^q_N: B_r^m \rightarrow B_r^m$ なる単項演算として扱い、 F^p_o に対応する

新演算系の二項演算は $F^p_N: B_r^m \times B_r^m \rightarrow B_r^m$ なる二項演算として扱い、 H^s_o に

- 5 対応する新演算系の T 項演算は $H^s_N: B_r^m \times B_r^m \times \cdots \times B_r^m \rightarrow B_r^m$ なる T 項演算として扱うことで、旧演算系の全ての演算と新演算系の全ての演算および符号器、復号器を全て多入力多出力の r 値論理の写像に対応させ、

B_r^n 上の任意 X に対応して、

$$(1c) \quad \chi'_{[X]}(\Phi(X)) = 1 \quad (\text{for } \forall X \in B_r^n)$$

10 $(2c) \quad \chi_X(\Psi([X])) = 1 \quad (\text{for } \forall X \in B_r^n)$

なる条件を満たす符号 $[X]$ ($[X] \subset B_r^m$) を生成するステップと、

新演算系の演算を生成するステップと、

上記演算のうち、

$$(3c) \quad \chi'_{[G^q_o(X)]}(G^q_N([X])) = 1 \quad (\text{for } \forall X \in B_r^n, \forall q)$$

15 $(4c) \quad \chi'_{[F^p_o(X,Y)]}(F^p_N([X],[Y])) = 1 \quad (\text{for } \forall X, Y \in B_r^n, \forall p)$

$$(5c) \quad \chi'_{[H^s_o(X_1, \dots, X_T)]}(H^s_N([X_1], \dots, [X_T])) = 1 \quad (\text{for } \forall X_1, \dots, X_T \in B_r^n, \forall s)$$

なる条件を満たす演算を選択するステップとをコンピュータに実行させることを特徴とする演算処理装置設計方法。

- 20 5. 有限集合を B^n 上の集合として扱い、

(6) B^n の任意の部分集合 S の特徴関数 $\chi_S(X)$ ($S \subseteq B^n$, $X \in B^n$) を1つの n 変数ブール関数 (特徴論理関数) とし、

(7) 集合の各要素を B^n を最小項に対応させ、

B^n の部分集合 S ($S \subset B^n$) の特徴論理関数を $\chi_S(X)$ ($X \in B^n$) と表し、また B^m

- 25 の部分集合 T ($T \subset B^m$) の特徴論理関数を $\chi'_T(Y)$ ($Y \in B^m$) と表し、 B^n の部分

集合 S の写像 $F : B^n \rightarrow B^m$ による像を $F(S)$ と表す時、

$$(8) \quad \chi_S(X) = \bigcup_{Q \in S} \chi_Q(X) = \bigcup_{Q \in S} X^Q$$

$$(9) \quad \chi'_{F(S)}(Y) = \bigcup_{Q \in S} Y^{F(Q)}$$

$$(10) \quad \chi'_{F(S)}(Y) = \bigcup_X Y^{F(X)} \cdot \chi_S(X)$$

5 なる関係式が成り立ち、さらに B^n の部分集合 $S, T \subset B^n$ に対して

$$(11) \quad \chi_S(X) \cdot \chi_T(X) = 0 \quad (\text{for } \forall X \in B^n) \Leftrightarrow S \cap T = \phi$$

$$(12) \quad \chi_{S \cap T}(X) = \chi_S(X) \cdot \chi_T(X)$$

$$(13) \quad \chi_{S \cup T}(X) = \chi_S(X) \cup \chi_T(X)$$

$$(14) \quad \overline{\chi_S(X)} \cdot \chi_T(X) = 0 \quad (\text{for } \forall X \in B^n) \Leftrightarrow S \supset T$$

10 なる関係式が成り立つことを利用し、

集合間の関係を入力するステップと、

入力された集合間の関係を関係式 (8) ~ (14) を用いて特徴論理関数の式に変換するステップと、

変換された特徴論理関数の式が成立するかどうかを判別するステップとをコン

15 ピュータに実行させることを特徴とする論理関数設計方法。

6. 2 値ブール代数 $B = \{0, 1\}$ 上に定義された m 種類の n 変数のブール関数 $f_j(X)$ (ここに $X = (x_1, x_2, \dots, x_n)$, $j=1, 2, \dots, m$) からなる写像 $F : B^n \rightarrow B^m$ を

$$20 \quad (16) \quad \tilde{F}(Y, X) = Y^{F(X)} = \bigcap_{j=1}^m \{y_j f_j(X) \cup \overline{y_j} \overline{f_j(X)}\} \quad (\text{ここに } Y = (y_1, y_2, \dots, y_m))$$

として定義した母関数 $\tilde{F}(X, Y)$ によって写像全体を一括して扱い、

$Y \in B^n$ の任意の関数 $g(Y)$ に対して

$$(18) \bigcup_Y g(Y) \cdot \tilde{F}(Y, X) = g(F(X))$$

なる関係式が成り立つこと、また恒等変換 $I: B^n \rightarrow B^n$ に対して

$$(19) \bigcup_Y g(Y) \cdot \tilde{I}(Y, X) = g(X)$$

なる関係式が成り立つこと、また写像 F の成分関数 $f_j(X)$ と母関数の関係

$$5 \quad (20) f_j(X) = \bigcup_Y y_j \cdot \tilde{F}(Y, X)$$

$$(21) \overline{f_j(X)} = \bigcup_Y \overline{y_j} \cdot \tilde{F}(Y, X)$$

$$(27) \tilde{F}(Y, X) \cdot \tilde{F}(Z, X) = \tilde{F}(Y, X) \cdot \tilde{I}(Y, Z)$$

なる関係式が成り立つこと、また写像 $F: B^n \rightarrow B^m$ と写像 $G: B^m \rightarrow B^l$ の合成写像を $R: B^n \rightarrow B^l$ とすると、

$$10 \quad (15) Z^{R(X)} = \bigcup_Y Z^{G(Y)} \cdot Y^{F(X)}$$

$$(17) \tilde{R}(Z, X) = \bigcup_Y \tilde{G}(Z, Y) \cdot \tilde{F}(Y, X)$$

なる関係式が成り立つこと、また $\Omega = B^n$ と置くと前記第2項記載の特徴論理関数に対して

$$(22) \bigcup_Y \tilde{F}(Y, X) = \chi_\Omega(X) = 1$$

$$15 \quad (23) \bigcup_X \tilde{F}(Y, X) = \chi'_{F(\Omega)}(Y)$$

なる関係式が成り立つこと、また $\Omega = B^n$ 上の同型写像 $\Phi: \Omega \rightarrow \Omega$ に対して、逆写像を $\Phi^{-1}: \Omega \rightarrow \Omega$ とすると前記母関数の定義 (16) に従い

$$(24) \tilde{\Phi}^{-1}(X, Y) = \tilde{\Phi}(Y, X) \quad (\text{for } X, Y \in \Omega)$$

$$(25) \phi^{-1}_i(X) = \bigcup_Y y_i \cdot \tilde{\Phi}(X, Y)$$

$$(26) \bigcup_Z \tilde{\Phi}^{-1}(X, Z) \cdot \tilde{\Phi}(Z, Y) = \bigcup_Z \tilde{\Phi}(X, Z) \cdot \tilde{\Phi}(Z, Y) = \tilde{I}(X, Y) \quad (\text{for } \forall X, Y, Z \in \Omega)$$

なる関係式が成り立つことを、また $Y \in B^n$ の任意の関数 $g(Y)$ に対して、

$$(28) \bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot g(Y) = \overline{g(Z)}$$

なる関係式が成り立つことで $g(Y)$ が最小項であることを判定し、また

$$5 \quad (29) \bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot \tilde{F}(Y, X) = \overline{\tilde{F}(Z, X)} \quad (\text{for } \forall Y, Z \in B^m, \forall X \in B^n)$$

なる関係式が成り立つことで $\tilde{F}(Y, X)$ が $B^n \rightarrow B^m$ なる写像の母関数であることを判定できることなどを利用する論理関数設計方法であって、

各写像の各成分関数または／および母関数を入力するステップと、

10 (16) 式に基づき母関数を計算するステップまたは／および (20) 式に基づき成分関数を計算するステップと、

(18) ～ (29) 式を計算するステップとをコンピュータに実行させ、もって、写像に関する処理や判定を成分毎に行うのではなく一括して扱うことを特徴とする論理関数設計方法。

15 7. 論理関数 $f(X)$ に対して

$$(30) f(X) \cdot \overline{f(Y)} \cdot \Delta(X, Y) = 0$$

なる関係式が成り立つかどうかを判定し、また写像 $F: B^n \rightarrow B^m$ の各成分関数 $f_j(X)$ ($j=1, 2, \dots, m$) に対しては写像の母関数を用い、

$$(31) \tilde{F}(X, A) \cdot \tilde{F}(Y, B) \cdot x_j \cdot \overline{y_j} \cdot \Delta_j(A, B) = 0$$

20 なる関係式が成り立つかどうかを判定し、さらに具体的には前記 (30) における $\Delta(X, Y)$ が $\Delta(X, Y) = \tilde{I}(X, Y^i)$ であること相当する

$$(32) f(X) \cdot \overline{f(Y)} \cdot \tilde{I}(X, Y^i) = 0$$

なる関係式、または $\Delta(X, Y) = \tilde{I}(X, Y^L)$ (for $\forall L \subset \Theta$) であることに相当する

$$(33) f(X) \cdot \overline{f(Y)} \cdot \tilde{I}(X, Y^L) = 0 \quad (\text{for } \forall L \subset \Theta)$$

なる関係式、または $\Delta(X, Y) = \bigcap_i (x_i \cdot y_i \cup \overline{x_i} \cdot \overline{y_i} \cup \overline{\theta^i})$ であることに相当する

$$(34) \quad f(X) \cdot \overline{f(Y)} \cdot \bigcap_i (x_i \cdot y_i \cup \overline{x_i} \cdot \overline{y_i} \cup \overline{\theta^i}) = 0$$

なる関係式、また前記 (31) における $\Delta_j(A, B)$ が

$$\Delta_j(A, B) = \bigcap_i (a_i \cdot b_i \cup \overline{a_i} \cdot \overline{b_i} \cup \theta_j^i)$$

$$5 \quad (35) \quad \tilde{F}(X, A) \cdot \tilde{F}(Y, B) \cdot x_j \cdot \overline{y_j} \cdot \bigcap_i (a_i \cdot b_i \cup \overline{a_i} \cdot \overline{b_i} \cup \theta_j^i) = 0$$

なる関係式を満足するかどうかを判定することで $f(X)$ の変数 x_1, x_2, \dots, x_n との依存関係を解析するでき、また母関数を用いることで各 $f_j(X)$ の変数 x_1, x_2, \dots, x_n との依存関係を一括して解析することができることを利用する論理関数設計方法であって、

- 10 写像の各成分関数または／および写像の母関数を入力するステップと、
 (30) ～ (35) 式に基づき変数依存度を解析するステップとをコンピュータに実行させることを特徴する論理関数設計方法。

8. 集合間の関係を入力するステップと、

- 15 入力された集合間の関係を関係式 (8) ～ (14) を用いて特徴論理関数の式に変換するステップと、

変換された特徴論理関数の式が成立するかどうかを判別するステップと、
 各写像の各成分関数または／および母関数を入力するステップと、

- 20 (16) 式に基づき母関数を計算するステップまたは／および (20) 式に基づき成分関数を計算するステップと、

(18) ～ (29) 式を計算するステップと、

(30) ～ (35) 式に基づき変数依存度を解析するステップとコンピュータに実行させて計算および判定を包括的に行う請求項5乃至請求項7に記載の論理関数設計方法。

9. 前記基数 r を $r = 2$ とし、 $i = 1, 2, \dots, n$ 、 $X, Y, Z, X_1, X_2, \dots, X_T \in B^n$ 、
 $j = 1, 2, \dots, m$ 、 $X', Y', Z', X'_1, X'_2, \dots, X'_T \in B^m$ とした時、旧演算系の各演算および新
 演算系の各演算、符号器、復号器の成分関数を、

$G^q_o : g_i^q_o(X)$ 、 $F^p_o : f_i^p_o(X, Y)$ 、 $H^s_o : h_i^s_o(X_1, X_2, \dots, X_T)$ および $G^q_N : g_j^q_N(X')$ 、

5 $F^p_N : f_j^p_N(X', Y')$ 、 $H^s_N : h_j^s_N(X'_1, X'_2, \dots, X'_T)$ 、 $\Phi : \phi_j(X)$ 、 $\Psi : \psi_i(X')$

と表し、さらに符号領域 C の特徴論理関数 $\chi_c(X')$ を $c(X')$ と表す時、

$$(1 \cdot b) \quad \tilde{\Psi}(X, X') \cdot c(X') \cdot \tilde{\Phi}(X', X) = \tilde{\Phi}(X', X)$$

$$(2 \cdot b) \quad \bigcup_{X'} \tilde{\Phi}(X', X) \cdot \tilde{\Phi}(X', Y) = \tilde{I}(X, Y)$$

$$(3 \cdot b \cdot 1) \quad \overline{\tilde{G}^q_o(Y, X)} \cdot \tilde{G}^q_N(Y', X') \cdot \tilde{\Psi}(Y, Y')c(Y') \cdot \tilde{\Psi}(X, X')c(X') = 0$$

$$10 \quad (3 \cdot b \cdot 2) \quad \overline{c(Y')} \cdot c(X') \cdot \tilde{G}^q_N(Y', X') = 0$$

$$(4 \cdot b \cdot 1)$$

$$\overline{\tilde{F}^p_o(Z, Y, X)} \cdot \tilde{F}^p_N(Z', Y', X') \cdot \tilde{\Psi}(Z, Z')c(Z') \cdot \tilde{\Psi}(Y, Y')c(Y') \cdot \tilde{\Psi}(X, X')c(X') = 0$$

$$(4 \cdot b \cdot 2) \quad \overline{c(Z')} \cdot c(Y') \cdot c(X') \cdot \tilde{F}^p_N(Z', Y', X') = 0$$

$$(5 \cdot b \cdot 1)$$

$$15 \quad \overline{\tilde{H}^s_o(Y, X_1, \dots, X_T)} \cdot \tilde{H}^s_N(Y', X'_1, \dots, X'_T) \cdot \tilde{\Psi}(Y, Y')c(Y') \cdot \tilde{\Psi}(X_1, X'_1)c(X'_1) \cdots \tilde{\Psi}(X_T, X'_T)c(X'_T) = 0$$

$$(5 \cdot b \cdot 2) \quad \overline{c(Y')} \cdot c(X'_1) \cdots c(X'_T) \cdot \tilde{H}^s_N(Y', X'_1, \dots, X'_T) = 0$$

(1-b) 乃至 (5-b-2) を符号化条件として用いる請求項 3 または 4 記載の演算
 処理装置設計方法。

20 10. 前記 (1-b) 乃至 (5-b-2) に加え、新演算系の演算器に対する論理式の
 簡単化の条件を付加し、その下で符号器、復号器、新演算器を論理設計すること
 で、新演算器の回路を簡単にする請求項 9 記載の演算処理装置設計方法。

11. 単項演算に対しては

$$25 \quad (48) \quad \tilde{G}^q_N(X', A') \cdot \tilde{G}^q_N(Y', B') \cdot x_j \cdot \overline{y_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\lambda_j^{q_j}}) = 0$$

なる条件を課し、二項演算に対しては

$$(49) \quad \tilde{F}^p_N(X', A', C') \cdot \tilde{F}^p_N(Y', B', C') \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\theta_1^{p_j i}}) = 0$$

$$\tilde{F}^p_N(X', C', A') \cdot \tilde{F}^p_N(Y', C', B') \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\theta_2^{p_j i}}) = 0$$

なる条件を課し、T 項演算に対しては

$$(50)$$

5

$$\tilde{H}^s_N(X', A', C'_2, \dots, C'_T) \cdot \tilde{H}^s_N(Y', B', C'_2, \dots, C'_T) \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\mu_1^{s_j i}}) = 0$$

$$\tilde{H}^s_N(X', C'_1, A', \dots, C'_T) \cdot \tilde{H}^s_N(Y', C'_1, B', \dots, C'_T) \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\mu_2^{s_j i}}) = 0$$

.....

$$\tilde{H}^s_N(X', C'_1, \dots, C'_{T-1}, A') \cdot \tilde{H}^s_N(Y', C'_1, \dots, C'_{T-1}, B') \cdot x'_j \cdot \overline{y'_j} \cdot \bigcap_i (a'_i \cdot b'_i \cup \overline{a'_i} \cdot \overline{b'_i} \cup \overline{\mu_T^{s_j i}}) = 0$$

- 10 なる変数依存条件を課し、新演算器の入力変数に対する変数依存度を旧演算系に比較し小さくなるように $\lambda^{q_j i}$ および $\theta_1^{p_j i}$ 、 $\theta_2^{p_j i}$ および $\mu_1^{s_j i}$ 乃至 $\mu_T^{s_j i}$ の値を定め、符号器、復号器、新演算器を論理設計することで、新演算器の回路規模および演算遅延時間を小さくする第 10 記載の演算処理装置設計方法。

- 15 12. 新演算系の二項演算に対しては前記 $\theta_1^{p_j i}$ と $\theta_2^{p_j i}$ に対して $\theta_1^{p_j i} = \theta_2^{p_j i}$ なる条件を課し、また新演算系の T 項演算に対しては前記 $\mu_1^{s_j i}$ 乃至 $\mu_T^{s_j i}$ に対して $\mu_1^{s_j i} = \dots = \mu_T^{s_j i}$ なる条件を課し、それぞれの演算を対称型で設計する請求項 9 記載の演算処理装置設計方法。

- 20 13. 前記旧表現データの空間 B_r^n と新表現データの空間 B_r^m を同一の空間 ($B_r^n = B_r^m, n = m$) にし、符号器 Φ を $\Phi: B_r^n \rightarrow B_r^n$ なる同型写像とし、復号器 Ψ を $\Psi = \Phi^{-1}$ (Φ の逆写像) とする請求項 3 または 4 記載の演算処理装置設計方法。

14.

$$(5.1) \quad \bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot \tilde{\Phi}(Y, X) = \overline{\tilde{\Phi}(Z, X)}$$

$$\bigcup_Y \overline{\tilde{I}(Z, Y)} \cdot \tilde{\Phi}(X, Y) = \overline{\tilde{\Phi}(X, Z)}$$

(5.5)

$$\tilde{G}^q_o(X, A) \cdot \tilde{G}^q_o(Y, B) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_I (\phi_I(A) \cdot \phi_I(B) \cup \overline{\phi_I(A)} \cdot \overline{\phi_I(B)} \cup \overline{\lambda^q_{j^I}}) = 0$$

(5.6)

$$5 \quad \tilde{F}^p_o(X, A, C) \cdot \tilde{F}^p_o(Y, B, C) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_I (\phi_I(A) \cdot \phi_I(B) \cup \overline{\phi_I(A)} \cdot \overline{\phi_I(B)} \cup \overline{\theta_1^p_{j^I}}) = 0$$

$$\tilde{F}^p_o(X, C, A) \cdot \tilde{F}^p_o(Y, C, B) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_I (\phi_I(A) \cdot \phi_I(B) \cup \overline{\phi_I(A)} \cdot \overline{\phi_I(B)} \cup \overline{\theta_2^p_{j^I}}) = 0$$

(5.7)

$$\tilde{H}^s_o(X, A, C_2, \dots, C_T) \cdot \tilde{H}^s_o(Y, B, C_2, \dots, C_T) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_I (\phi_I(A) \cdot \phi_I(B) \cup \overline{\phi_I(A)} \cdot \overline{\phi_I(B)} \cup \overline{\mu_1^s_{j^I}}) = 0$$

$$\tilde{H}^s_o(X, C_1, A, \dots, C_T) \cdot \tilde{H}^s_o(Y, C_1, B, \dots, C_T) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_I (\phi_I(A) \cdot \phi_I(B) \cup \overline{\phi_I(A)} \cdot \overline{\phi_I(B)} \cup \overline{\mu_2^s_{j^I}}) = 0$$

10

.

$$\tilde{H}^s_o(X, C_1, C_2, \dots, A) \cdot \tilde{H}^s_o(Y, C_1, C_2, \dots, B) \cdot \phi_j(X) \cdot \overline{\phi_j(Y)} \cdot \bigcap_I (\phi_I(A) \cdot \phi_I(B) \cup \overline{\phi_I(A)} \cdot \overline{\phi_I(B)} \cup \overline{\mu_T^s_{j^I}}) = 0$$

上記 (5.1) および (5.5) 乃至 (5.7) 式を満足するように符号器 Φ を決定し、

復号器 Ψ は $\Psi = \Phi^{-1}$ として決定し、

$$(5.2) \quad \tilde{G}^q_N(Y', X') = \bigcup_Y \bigcup_X \tilde{G}^q_o(Y, X) \cdot \tilde{\Phi}(Y', Y) \cdot \tilde{\Phi}(X', X)$$

15

$$(5.3) \quad \tilde{F}^p_N(Z', X', Y') = \bigcup_Z \bigcup_X \bigcup_Y \tilde{F}^p_o(Z, X, Y) \cdot \tilde{\Phi}(Z', Z) \cdot \tilde{\Phi}(X', X) \cdot \tilde{\Phi}(Y', Y)$$

$$(5.4) \quad \tilde{H}^s_N(Y', X'_1, \dots, X'_T) = \bigcup_Y \bigcup_{X_1} \dots \bigcup_{X_T} \tilde{H}^s_o(Y, X_1, \dots, X_T) \cdot \tilde{\Phi}(Y', Y) \cdot \tilde{\Phi}(X'_1, X_1) \cdot \dots \cdot \tilde{\Phi}(X'_T, X_T)$$

上記（５２）乃至（５４）より各新演算系に対する変数依存度 λ_j^i および θ_1^{p,j^i} 、 θ_2^{p,j^i} および μ_1^{s,j^i} 乃至 μ_T^{s,j^i} の下に新演算系の演算器を決定する請求項１３記載の演算処理装置設計方法。

要約書

純粹に論理的なアプローチにより、演算回路の素子数および遅延時間を削減した演算装置を実現するため、符号化による演算方式を具体的且つ効率的に論理設計し、符号化演算装置を提供する。演算系の演算器を必要があれば拡張し、基数 r の論理関数として扱い、また $r = 2$ の時には写像の新たに表現である母関数を用い、符号化条件および論理式簡単化条件の下に新演算系を論理設計するか、または元演算系と新演算系の演算器の入出力関係のトポロジーを一致させることにより符号化演算方式の新演算系を設計する。符号化条件および論理式簡単化条件を満たす演算処理装置は、高速化・低消費電力を実現する。

1/18

図1

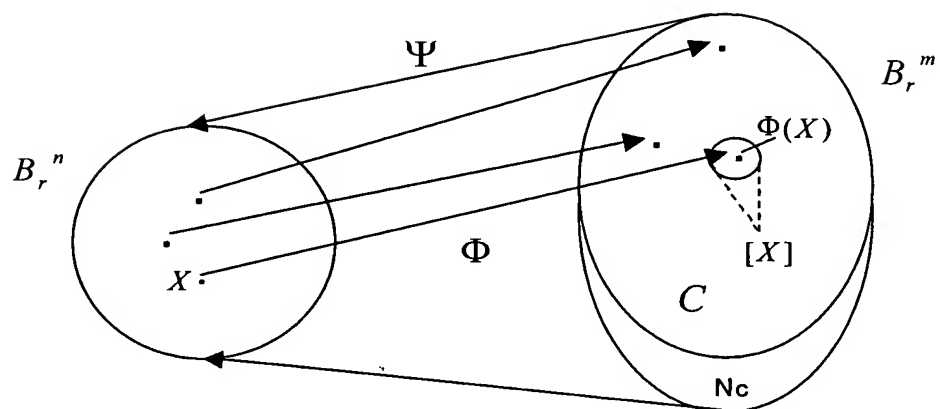
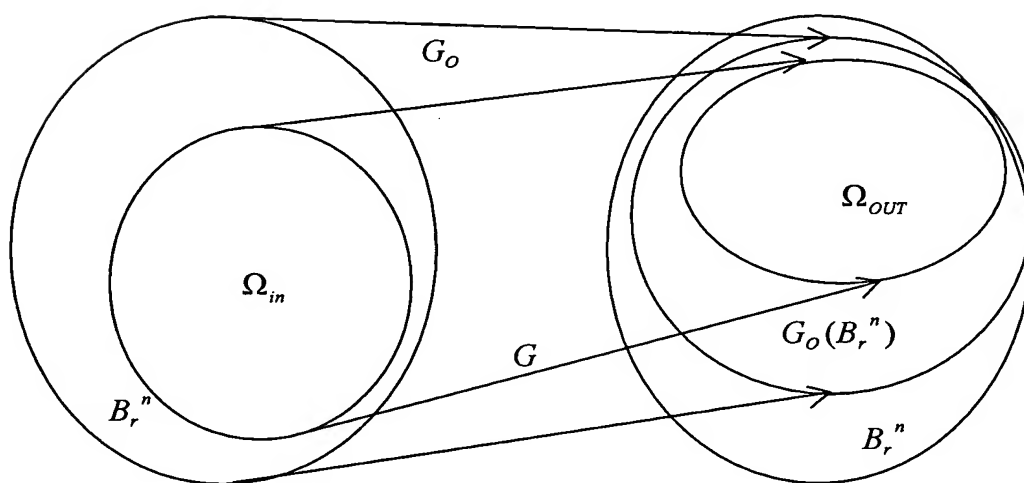
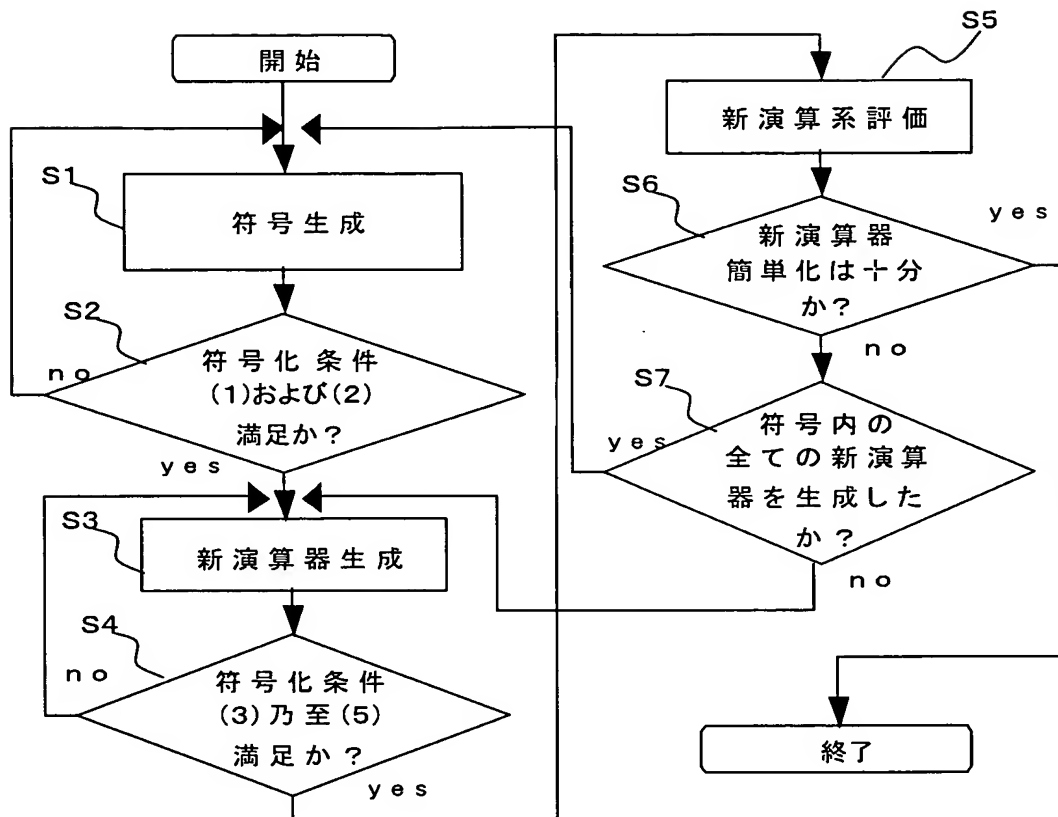


図2



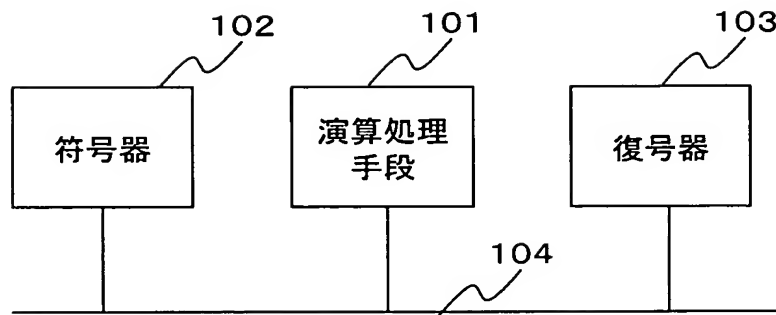
2/18

図3



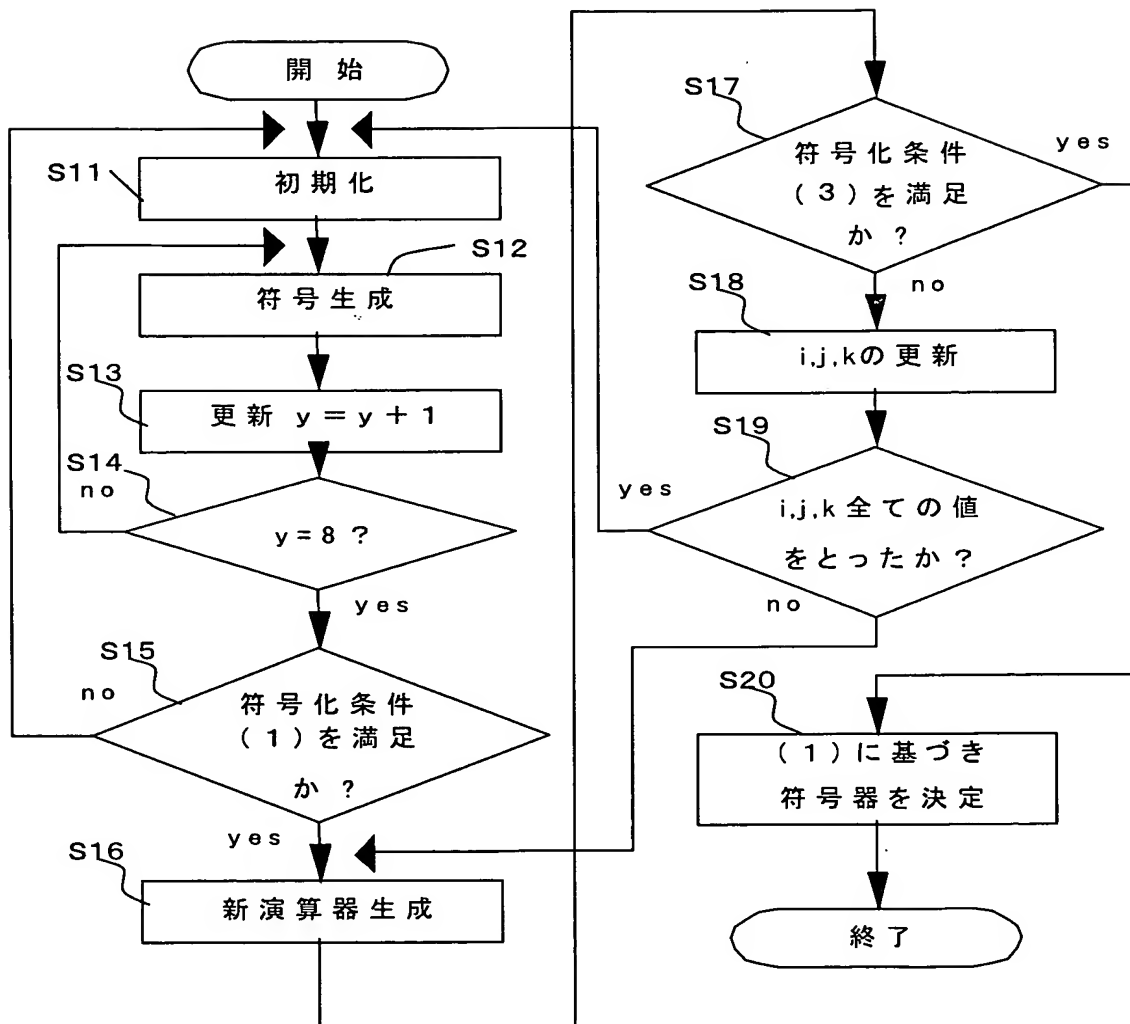
3/18

図4

演算処理装置100

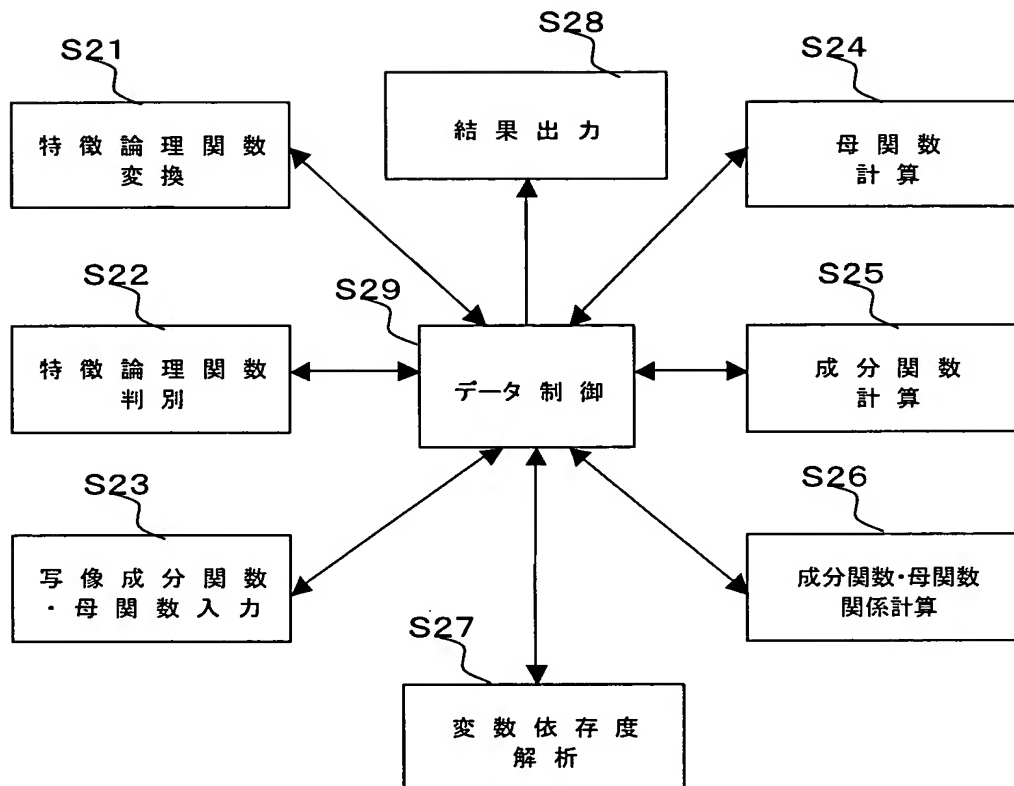
4/18

図5



5/18

図6



6/18

图7

$$Z' = F^1(X, Y)$$

		Y			
		00	01	10	11
X	00	011	110	000	101
	01	110	111	110	110
	10	000	110	000	111
	11	101	110	111	101

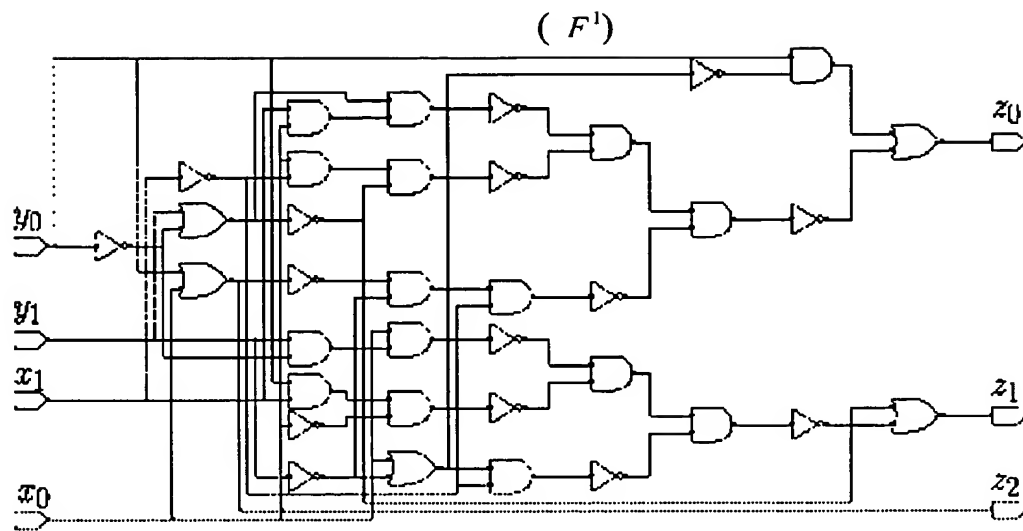
图8

$$Z' = F^2(X', Y')$$

		Y'							
		000	001	010	011	100	101	110	111
X'	000	010	001	110	100	001	100	110	010
	001	001	011	100	001	100	101	011	101
	010	110	100	010	001	100	001	010	110
	011	100	001	001	100	001	100	001	100
	100	001	100	100	001	100	001	100	001
	101	100	101	001	100	001	011	101	011
	110	110	011	010	001	100	101	000	111
	111	010	101	110	100	001	011	111	000

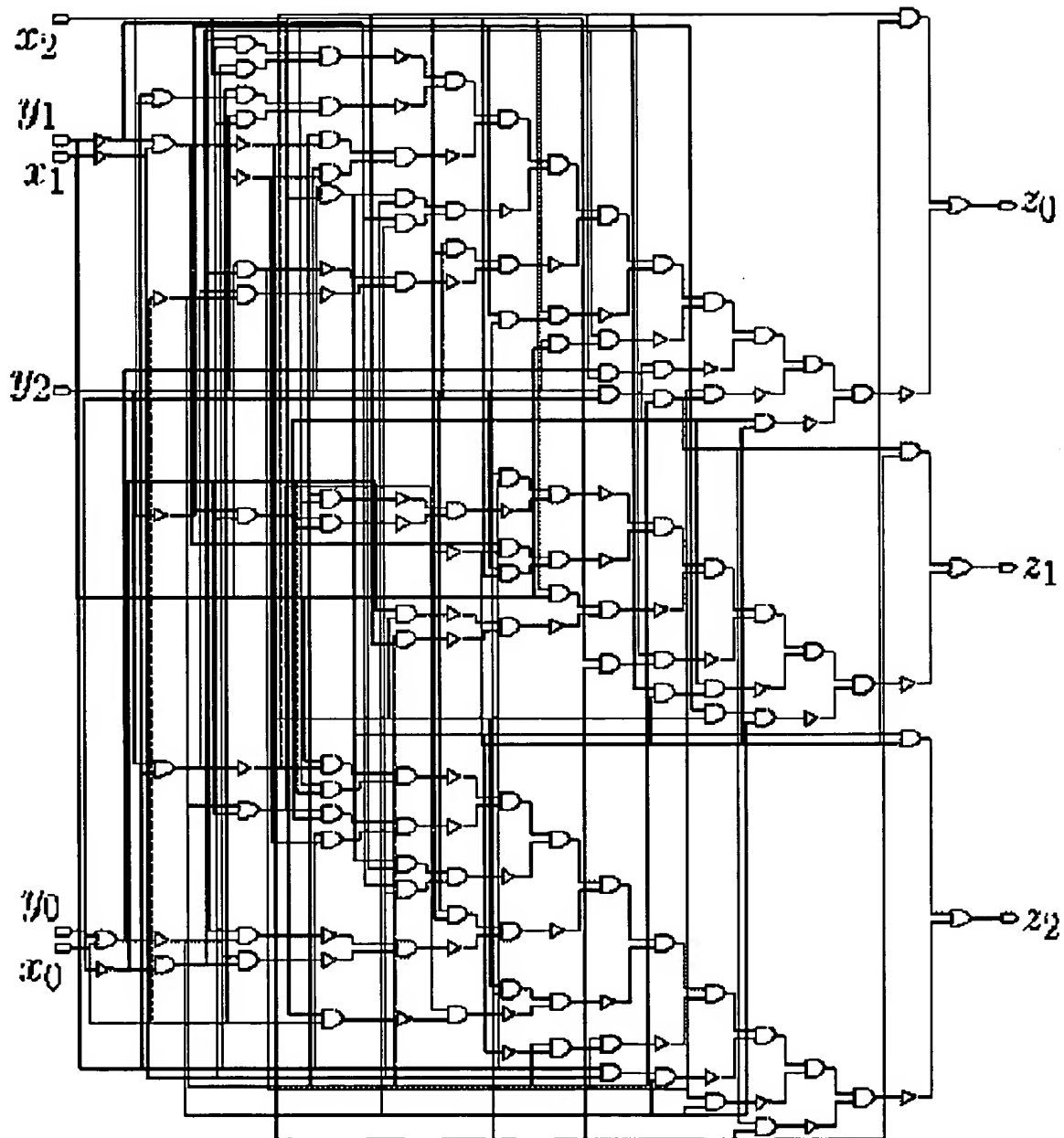
7/18

図9



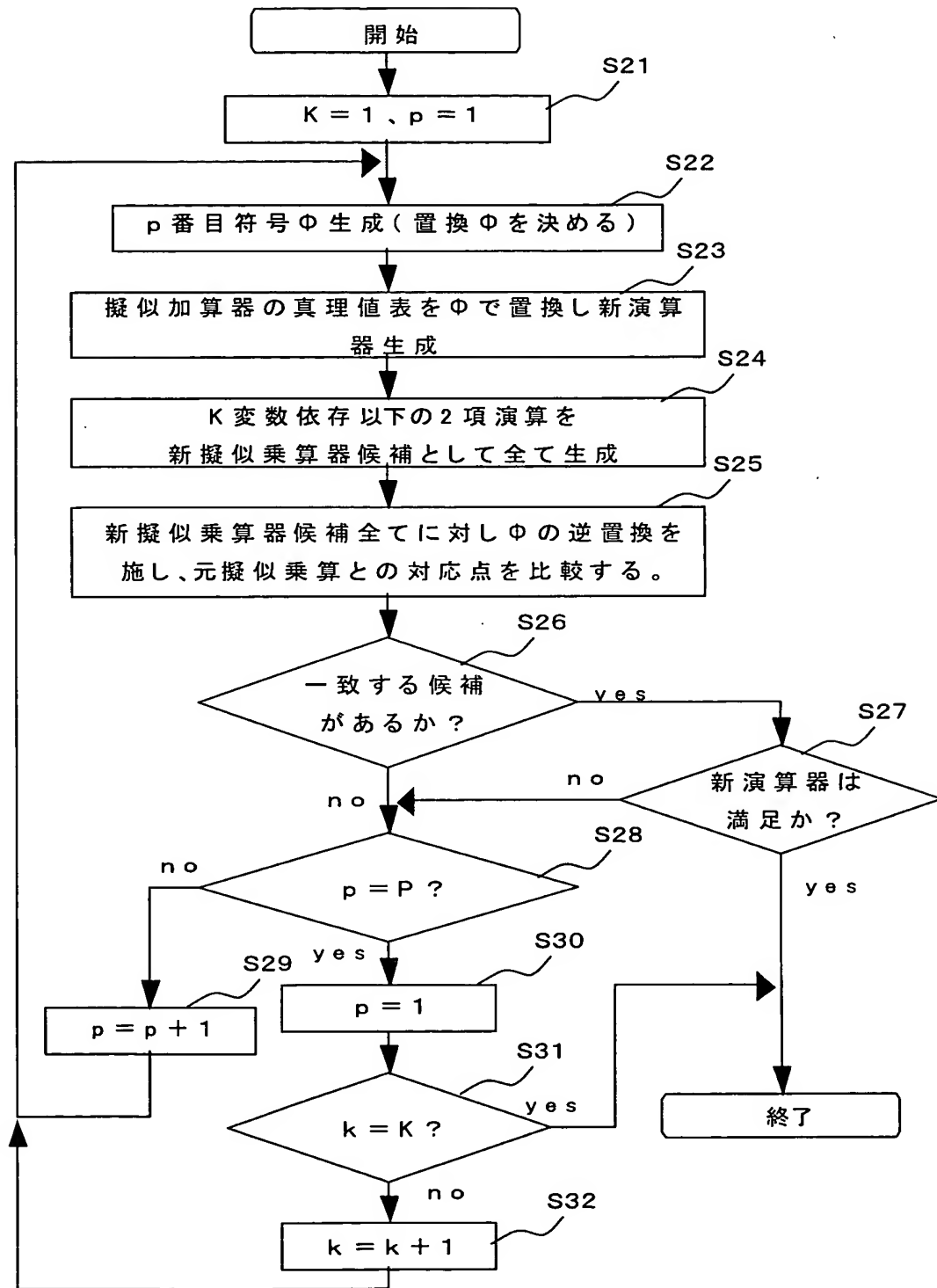
8/18

図 10

 (F^2) 

9/18

図11



10/18

図12

$$Z' = F^{-1}_N(X', Y')$$

		Y'							
		000	001	010	011	100	101	110	111
X'	000	000	100	000	100	010	110	010	110
	001	100	000	100	000	110	010	110	010
	010	000	100	001	101	010	110	011	111
	011	100	000	101	001	110	010	111	011
	100	010	110	010	110	010	110	010	110
	101	110	010	110	010	110	010	110	010
	110	010	110	011	111	010	110	011	111
	111	110	010	111	011	110	010	111	011

図13

		$\Phi(X')$
X	000	011
	001	100
	010	111
	011	001
	100	101
	101	000
	110	110
	111	010

11/18

图 14

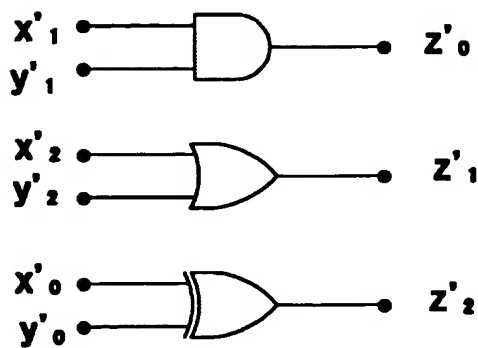
$$Z' = F^2_N(X', Y')$$

		000	001	010	011	100	101	110	111
x'	000	001	101	001	101	000	100	000	100
	001	101	101	101	101	100	100	100	100
	010	001	101	011	111	000	100	010	110
	011	100	000	101	001	110	010	111	011
	100	101	101	111	111	100	100	110	110
	101	100	100	100	100	101	101	101	101
	110	000	100	010	110	001	101	011	111
	111	100	100	110	110	101	101	111	111

图 15

	$\Psi(X')$
000	101
001	011
010	111
011	001
100	001
101	100
110	110
111	010

图 16



12/18

图 17

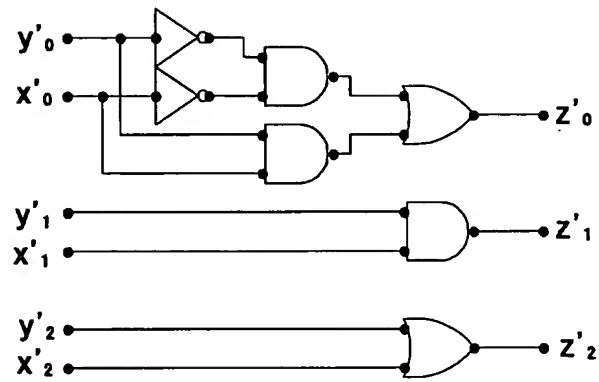
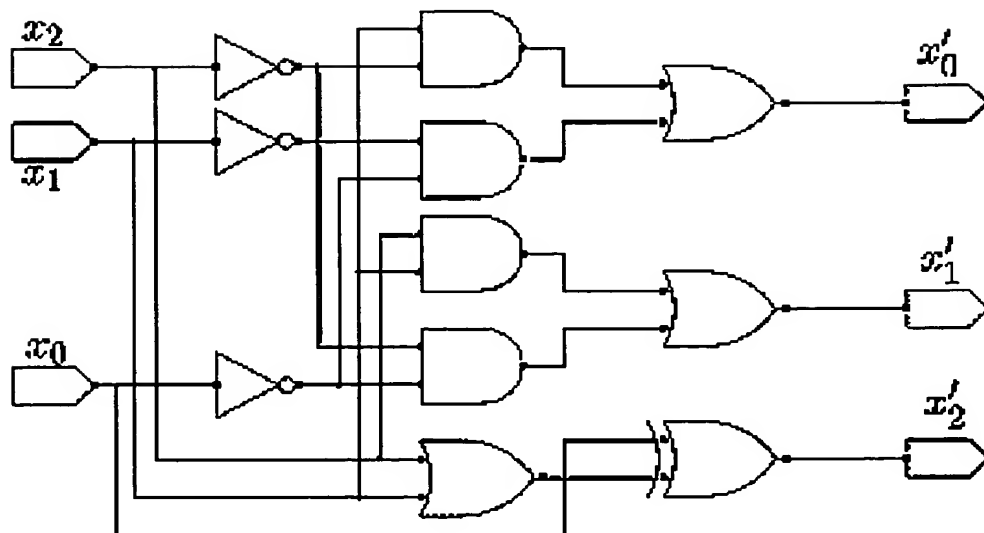


图 18

(Φ)

13/18

図19

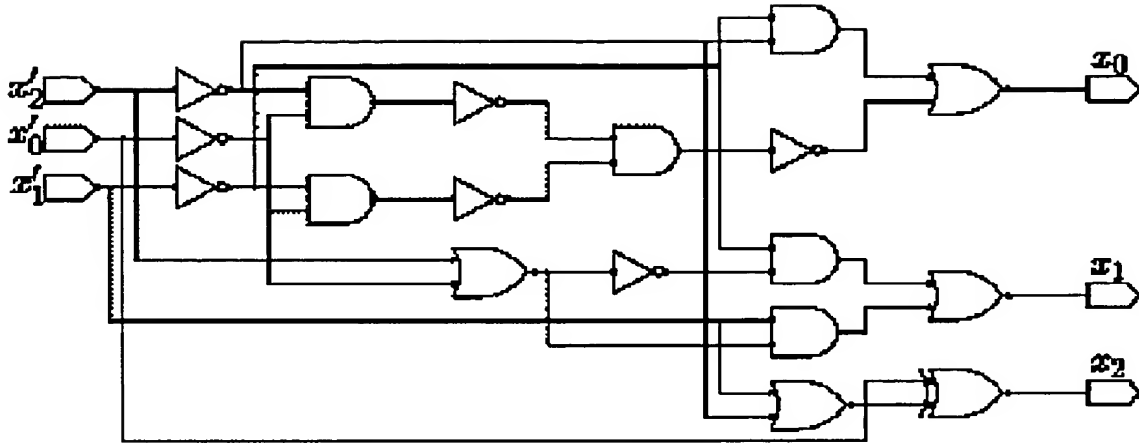
(Ψ)

図20

		素子数	遅延時間
元演算系	F^1	$\square^1 = 57\square$	$\triangle^1 = 13\triangle$
	F^2	$\square^2 = 267\square$	$\triangle^2 = 29\triangle$
新演算系	F^1_N	$\square^1_N = 2\square$	$\triangle^1_N = 6\triangle$
	F^2_N	$\square^2_N = 12\square$	$\triangle^2_N = 5\triangle$
	Φ	$\square_\Phi = 21\square$	$\triangle_\Phi = 5\triangle$
	Ψ	$\square_\Psi = 31\square$	$\triangle_\Psi = 10\triangle$

图21

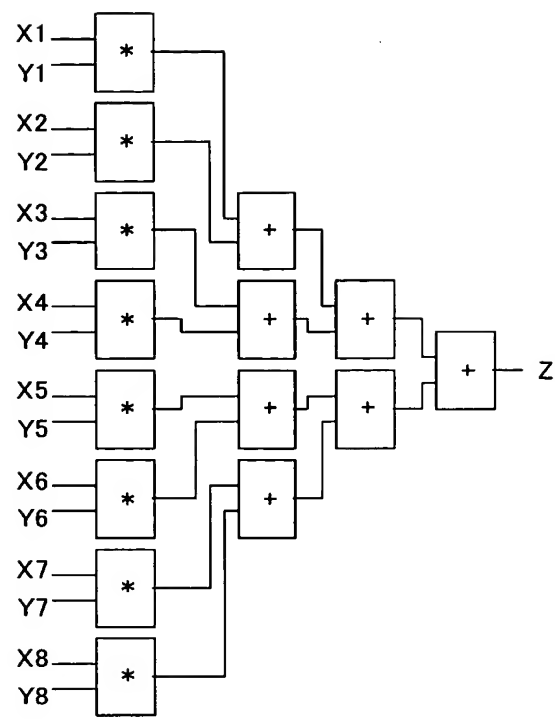
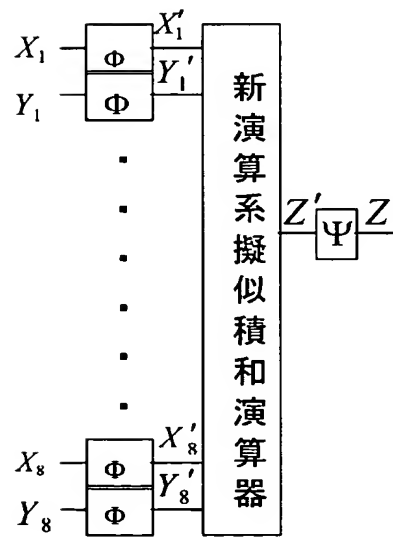


图22



15/18

図23

		$G^1(X)$			$G^2(X)$
X	00	00	X	00	00
	01	00		01	00
	10	01		10	01
	11	11		11	10

図24

$\tilde{G}^1(Y, X)$					$\tilde{G}^2(Y, X)$				
	X^0	X^1	X^2	X^3		X^0	X^1	X^2	X^3
Y^0	1	1	0	0	Y^0	1	1	0	0
Y^1	0	0	1	0	Y^1	0	0	1	0
Y^2	0	0	0	0	Y^2	0	0	0	1
Y^3	0	0	0	1	Y^3	0	0	0	0

16/18

図25

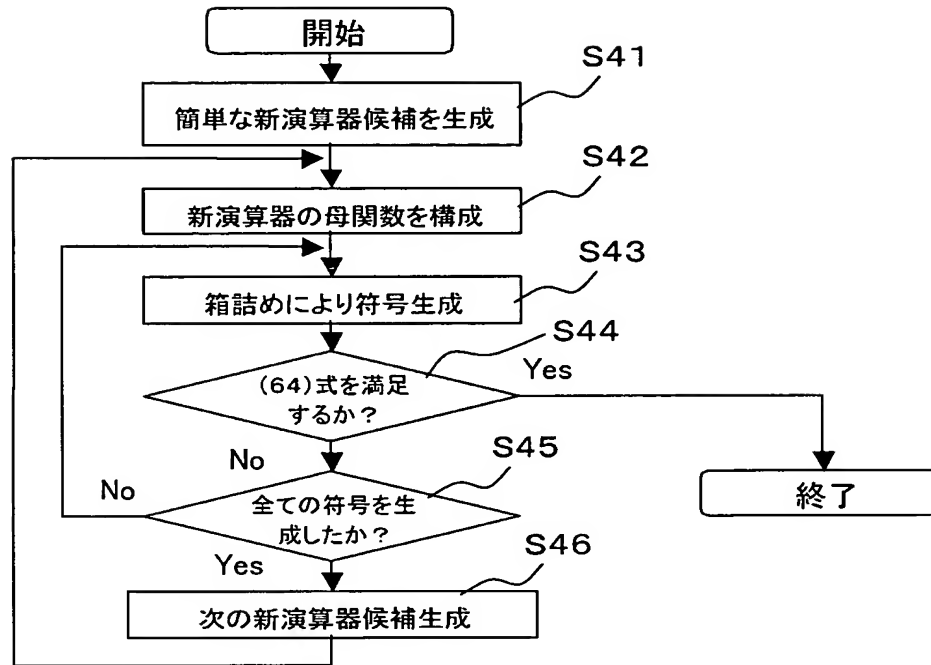


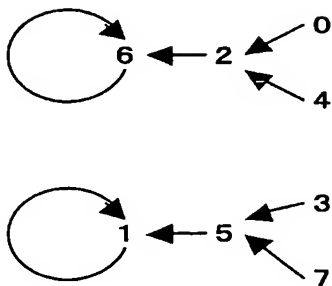
図26



17/18

図27

G^1_N の入出力ポロジ



G^2_N の入出力ポロジ

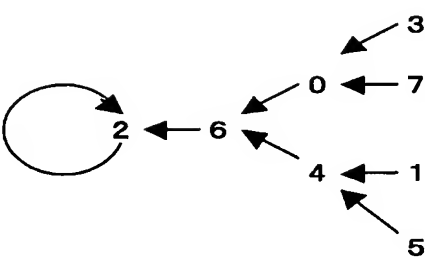
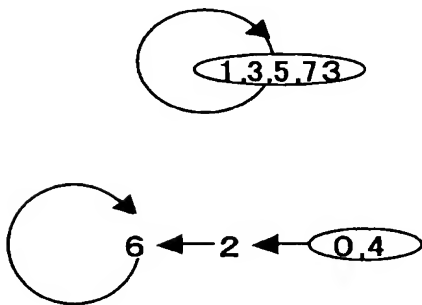
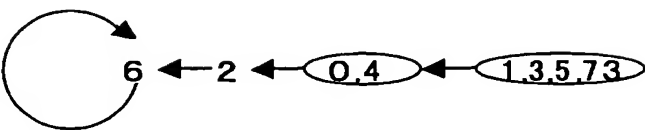


図28

G^1_N の束ねた入出力ポロジ



G^2_N の束ねた入出力ポロジ



18/18

図29

